Streamlining Missing Data Analysis by Aggregating Multiple Imputations at the Data Level: A Monte Carlo Simulation to Assess the Tenability of the SuperMatrix Approach

BY

Kyle M. Lang

Submitted to the graduate program in Psychology and
the Graduate Faculty of the University of Kansas
in partial fulfillment of the requirements for the degree of
Master of Arts

Chairperson Todd D. Little

Wei Wu

Paul E. Johnson

Date defended: February 13, 2013

The Thesis Committee for Kyle M. Lang certifies
that this is the approved version of the following thesis :

Streamlining Missing Data Analysis by Aggregating Multiple Imputations at the Data Level:
A Monte Carlo Simulation to Assess the Tenability of the SuperMatrix Approach

Chairperson Todd D. Little

Date approved: February 14, 2013

Abstract

A Monte Carlo Simulation Study was conducted to assess the tenability of a novel treatment of missing data. Through aggregating multiply-imputed data sets prior to model estimation, the proposed technique allows researchers to reap the benefits of a principled missing data tool (i.e., multiple imputation), while maintaining the simplicity of complete case analysis. In terms of the accuracy of model fit indices derived from confirmatory factor analyses, the proposed technique was found to perform universally better than a naive *ad hoc* technique consisting of averaging the multiple estimates of model fit derived from a traditionally conceived implementation of multiple imputation. However, the proposed technique performed considerably worse in this task than did full information maximum likelihood (FIML) estimation. Absolute fit indices and residual based fit indices derived from the proposed technique demonstrated an unacceptable degree of bias in assessing direct model fit, but incremental fit indices led to acceptable conclusions regarding model fit. $\Delta\chi^2$ values derived from the proposed technique were unbiased across all study conditions (except for those with very poor parameterizations) and were consistently more accurate than such values derived from the *ad hoc* comparison condition. It was also found that $\Delta\chi^2$ values derived from FIML-based models were negatively biased to an unacceptable degree in any conditions with greater than 10% missing. Implications, limitations and future directions of the current work are discussed.

Acknowledgements

I would like to thank my advisor, Dr. Todd Little. It was Dr. Little who originally conceptualized the SuperMatrix technique and gave me the opportunity to examine its performance. I am profoundly thankful for his continued patience and support as the current project has come to fruition. I also want to thank Dr. Paul Johnson for his enthusiastic assistance in designing and implementing the simulation from which the findings of this study are derived. Without Dr. Johnson's guidance, computational restrictions would have prevented the current work from achieving anywhere near its final scope. I am also indebted to Dr. Wei Wu for her invaluable input when working through the initial theoretical pitfalls of this project and for her detailed comments on the final draft of this paper. Especially in the early stages of this study, Dr. Wu's guidance and collaboration were crucial.

Additionally, I must acknowledge the considerable contributions of Dr. Alex Schoemann, Dr. Mijke Rhemtulla, Kimberly Gibson, Sunthud Pornprasertmanit, Mauricio Garnier Villarreal, Patrick Miller, Dr. Waylon Howard, Dr. Kris Preacher, Rawni Anderson, Dr. John Geldhof, and the entire CRMDA Missing Data Workgroup. Without the collaboration of this community of missing data experts, the current project would not have been possible.

Finally, I wish to thank my mother and father Lori and Scott Lang, my siblings Anthony, Dalton, and Maggie, and my grandmother Beverly Bareiss. I owe my current success as much to a lifetime of their undying patience and support as any other factor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Missing data are an omnipresent reality in the work of the social scientist and certainly not least in that of psychologists. Perhaps more so than any other scientific discipline, psychology is intimately tied, at its most fundamental level, with the most sensitive areas of the human condition. Therefore, any academician attempting to shed an inquiring light on the sheltered corners of the human psyche must necessarily probe their subjects' most private and personal beliefs—the type of inquiry to which these subjects are least likely to acquiesce. Add to this the inherently fickle nature of human subjects, and the prevalence of nonresponse in psychological studies tends to be quite high.

As this has been the nature of experimental psychology since its very inception, it is little surprise that advancements in missing data analysis are keenly sought after by so many researchers in the field. However, the unfortunate truth of the world is such that a technique which is difficult to implement correctly or onerously time consuming when applied (no matter how wonderful it may be) has a very small chance of becoming a widely utilized tool. Thus, any such overly burdensome advancement is rendered ineffectual in improving the quality of the average applied researcher's methodology and is subsequently stripped of nearly all power to advance the state of psychological science. It was a cognizance of this unfortunate reality that led the author and his colleagues to instigate the current project. The pursuit of an easily accessible, yet methodologically principled, solution to an all-too-common problem was the impetus for the current project.

Time and time again the author has run into the same frustrating conundrum. Namely, when fitting latent variable models to multiply-imputed data there is no readily apparent approach with which the *goodness-of-fit* of a hypothesized model can be determined. Customarily, after a run of any given multiple imputation algorithm, the researcher is left with m separate imputed data sets. The usual protocol would entail analyzing each of these m data sets separately and combining the resulting m sets of parameter estimate using *Rubin's Rules* (Rubin, 1987). However, Rubin's Rules were not developed with latent variable modeling in mind and are principally intended only to correct standard errors in order to minimize bias in the test statistics associated with model parameters. Therefore, they are not directly applicable to the aggregation of the χ^2 [1] (or χ^2-based model fit indices) across multiple imputations. Previous work has developed χ^2-distributed statistics which can be calculated in the context of multiply imputed data structures and offer unbiased assessments of model fit (e.g., Browne, 1984; Meng & Rubin, 1992). Unfortunately, these statistics tend to entail difficult computations that can make them difficult to implement conscientiously, and they have yet to be widely implemented in popular statistical analysis software.

Over the years, several solutions to this conundrum have been proposed and tested, but it is the opinion of the author that a solution that is sufficiently simple to become a widely utilized tool has yet to be suggested. Unfortunately, the simplest *ad hoc* solution to this problem has been shown to provide suboptimal results. Although intuitively appealing, simply averaging the m χ^2 statistics after fitting the analysis models will lead to biased assessments of model fit (Asparouhov & Muthén, 2010). More principled approaches that have been suggested in the literature include averaging the covariance matrices derived from multiple EM imputations (Cai & Lee, 2009) and averaging the m covariance matrices calculated from the m imputed data sets produced by a Bayesian multiple imputation (Lee & Cai, 2012). However, these approaches are both two-stage procedures that require subsequent *post hoc* correction to the resulting likelihood ratio (LR) statistic before it follows a χ^2 distribution. Additionally, neither of these papers explicitly addressed the performance

[1] It should be understood that the usage of the term χ^2, when referenced as a model fit index, constitutes a slight abuse of notation. The more appropriate name for this statistic would be *Likelihood Ratio Statistic*. However, these terms will be used interchangeably throughout this paper.

of their proposed estimators in the context of hypothesis testing[2].

Developments that are particularly germane to the current project have arisen in the context of variable selection for stepwise regression procedures. Wood, White, and Royston (2008) have suggested a "stacked data" approach based on aggregating the m imputed data sets by stacking them one atop the other into an $mN \times p$ data frame where N is the original number of observations and p is the number of variables. When this stacked data frame was subsequently analyzed with a series of weighted linear regression models, the resulting Type I error rates and power were comparable to those derived from Rubin's Rules, even with relatively simple weighting schemes. The current work develops a technique that is mathematically equivalent to the W_1 technique from Wood et al. (2008), yet offers a simplified implementation that is tailored to covariance structure modeling. Our approach was designed to aggregate the multiply imputed data sets prior to model estimation in a fashion that will allow researchers to address their missing data issues in a principled fashion (i.e., by leveraging the benefits of multiple imputation), yet still maintain the simplicity of estimating a single analysis model.

1.1 The Proposed Technique

As the desired product after the application of this technique is a condensed data structure which aggregates a number of multiply imputed data sets, the first step is creating some number of plausible imputations. Of course, all of the usual tenets underlying any missing data analysis remain, and all of the customary considerations must be made in constructing these imputations. The technique under study is not envisioned as a correction for any shortcomings in the initial imputation scheme, and all subsequent steps are conducted under the implicit assumption that the imputations upon which they are performed were well constructed (see Enders, 2010; Schafer & Graham, 2002 for very accessible overviews of multiple imputation and missing data analysis, and Rubin, 1987; Schafer, 1997 for more technical treatments of multiple imputation).

[2]While both of these papers do present the structural parameters and associated standard errors derived from applying their proposed estimators to example data, the correctness of the values is never thoroughly scrutinized.

Once the imputations have been created, the proposed technique is very simple to implement. After some number ($m > 1$) of imputations have been created, these m imputed data sets are "stacked" one atop the other to create a single data frame whose number of rows is equal to the number of observations in the original data set times the number of imputations. For example, consider a data set that contains $N = 250$ observations of $p = 25$ variables. If you create $m = 20$ imputations, the final dimensions of this combined data frame will be $mN \times p = 20(250) \times 25 = 5000 \times 25$. Once this aggregated data frame has been constructed, it is summarized with a single covariance matrix that is calculated from all of the mN observations. This covariance matrix then acts as sufficient statistics for any subsequent modeling. In the example above, this would lead to a 25×25 covariance matrix that would then be used as input for future analyses. Figure 1.1 gives a graphical representation of the process for a trivial example in which the original data are comprised of 4 observations of 3 variables.

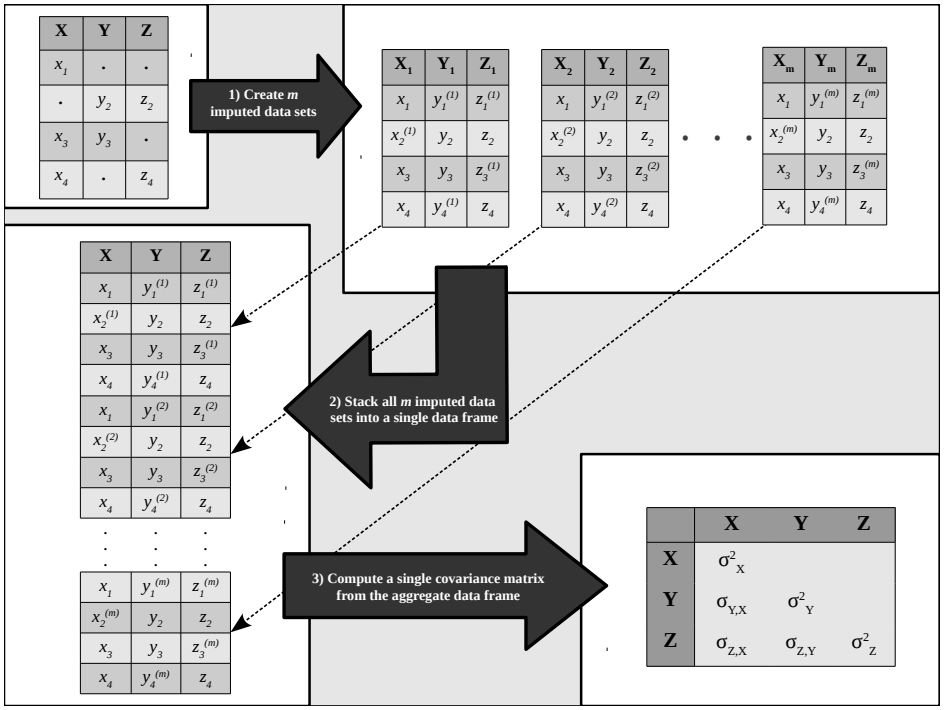

Figure 1.1: A graphic representation of the SuperMatrix technique as applied to a data set with 4 observations of 3 variables

Because the aggregate covariance matrix that is produced by this process is envisioned as a

proxy of the true *complete-data* covariance matrix, the number of observations ascribed to the underlying data set is explicitly specified to be equal to the original sample size N. This constraint has the effect of downwardly adjusting the LR statistics derived from these models to correct for the spurious inflation of sample size that follows from treating the stacked m imputed data sets as a single data frame. Incidentally, this correction also makes the proposed technique mathematically equivalent to the W_1 weighting scheme proposed by Wood et al. (2008).

In offering guidance to researchers seeking to implement multiple imputation techniques, van Buuren (2012) suggests that stacking techniques which are similar to the technique proposed here will produce unbiased point estimates of model parameters, yet may lead to negatively biased standard errors for those estimate. Likewise, our proposed technique (hereafter referred to as the *SuperMatrix* technique) is expected to produce unbiased point estimates of model parameters, but we do not have confidence that it will produce accurate standard errors[3]. However, it is intuited that the SuperMatrix (SM) technique will maintain enough of the total information inherent in the original m imputed data sets to produce, in the course of standard model fitting procedures, an acceptable assessment of model fit. Further, because significance tests that employ nested model $\Delta\chi^2$ statistics are not affected by the tested parameters' standard errors, we anticipate the SM technique can be used to achieve accurate tests of hypotheses when using the $\Delta\chi^2$ test within a model comparison framework.

This final point is considered to be of paramount importance because it appreciably extends the utility of the current work. While there are extant techniques that can produce unbiased Wald statistics in the context of multiply-imputed data structures (i.e., *Rubin's Rules*), these techniques are, in a certain sense, suboptimal. The problem with hypothesis testing through *Rubin's Rules* is essentially two-fold. First, standard errors can be highly sensitive to distributional assumptions and the specifics of model identification (even with fully observed data). Second, many of the hypotheses tested with latent variable models entail the comparison of alternative understandings of a phenomenon that are quantified as competing models (as opposed to hypotheses that involve testing

[3]The reader is encouraged to refer to Wood et al. (2008, Appendix A) to see an analytic justification for this expectation.

single structural parameters). Taking these two points under consideration, we see that it is often desirable to forgo single parameter tests entirely and implement a model-comparison paradigm that employs the $\Delta\chi^2$ test as its primary test statistic. However, as discussed above, χ^2 statistics are not naturally forthcoming in the context of multiply-imputed data structures. Thus, the SM technique is envisioned as not only offering an easily implemented method of assessing goodness-of-fit in latent variable models but also providing a very parsimonious method with which to test complex, multi-parameter hypotheses.

A Monte Carlo simulation study was conducted to test the tenability of the SM technique while varying sample size and percent missing. Of particular interest were convergence rates of the analysis models, assessments of direct model fit (i.e., the analysis models' goodness-of-fit to the simulated data), and the accuracy of hypothesis tests conducted with nested model $\Delta\chi^2$ statistics. Specifically, it is hypothesized that:

1. Across the replications of the Monte Carlo study, the SM conditions will show higher rates of convergence than will the conditions based on full information maximum likelihood (FIML) estimation[4]. This effect will be especially pronounced for lower sample sizes and higher percents missing.

2. Estimates of model fit derived from the SM technique will show negligible deviation from estimates of model fit derived from complete data comparison conditions. Although, this performance is expected to deteriorate monotonically as sample size decreases and percent missing increases.

3. Estimates of model fit derived from an *ad hoc* technique consisting of simply averaging the *m* estimates of model fit (hereafter referred to as the *Naive* approach) will show universally larger deviations from complete data-based estimates than will estimates derived from the SM technique.

[4]Although there are no specific hypotheses regarding the relative performance of the SM technique and FIML estimation when it comes to assessments of direct model fit, it is generally expected that the FIML conditions will outperform the SM conditions in this regard.

4. Across all conditions, a nested model χ^2 difference test conducted using the SM-based $\Delta\chi^2$ values will show negligible deviation from the same tests conducted with the complete data-based $\Delta\chi^2$ and will be universally closer to the complete data values than will such tests employing the Naive-based $\Delta\chi^2$.

Chapter 2

Methods

2.1 Data Simulation

The data for the Monte Carlo study were simulated using R 2.15[1] (R Development Core Team, 2011). The data for each observation were derived from a multitrait multimethod (MTMM) confirmatory factor analytic model with fixed, known population-level parameter values. This model consisted of four latent factors which acted as predictors of twenty manifest variables. Two of these factors were conceptualized as "common factors" underlying the twenty manifest variables and were each exclusively associated with ten of these variables. The third and fourth factors were conceptualized as "covariate factors." These factors were introduced into the model in the same fashion as method factors in a traditional MTMM model, except that both covariate factors were allowed to predict all twenty measured variables.

All "common factor" loadings were specified to be equal in the population model with a value of $\lambda = .6$ (i.e., the measurement model was essentially tau equivalent, at the population level). The covariate factors were each allowed to a have a different, small association with the two simulated "scales." That is, the first ten manifest variables loaded onto the first covariate factor at $\lambda = .2$ while the second ten loaded onto this factor at $\lambda = .1$. Likewise, the first ten manifest variables loaded

[1] Code excerpt 1 shows the source code of the function used to simulation the complete data.

onto the second covariate factor at $\lambda = .05$ while the second ten did so at $\lambda = .1$. The primary purpose of these covariate factors was to provide factor scores that would act as predictors of the missing data process. However, including them in the data simulation model, as well, increased the ecological validity of the results by conditioning the missingness on variables which also had a direct connection to the simulated data.

Common factors had variances fixed to $\psi_{1,1} = \psi_{2,2} = 1.0$ and were allowed to covary with one another at $\psi_{2,1} = .5$. The two covariate factors were specified to be independent of one another and the common factors. They had variances of $\psi_{3,3} = 1.0$ and $\psi_{4,4} = 3.0$. Variances of all unique factors were specified to be a constant $\theta = 1 - \lambda^2 = 1 - .6^2 = .64$. The final form of the population model is shown in Figure 2.1[2].

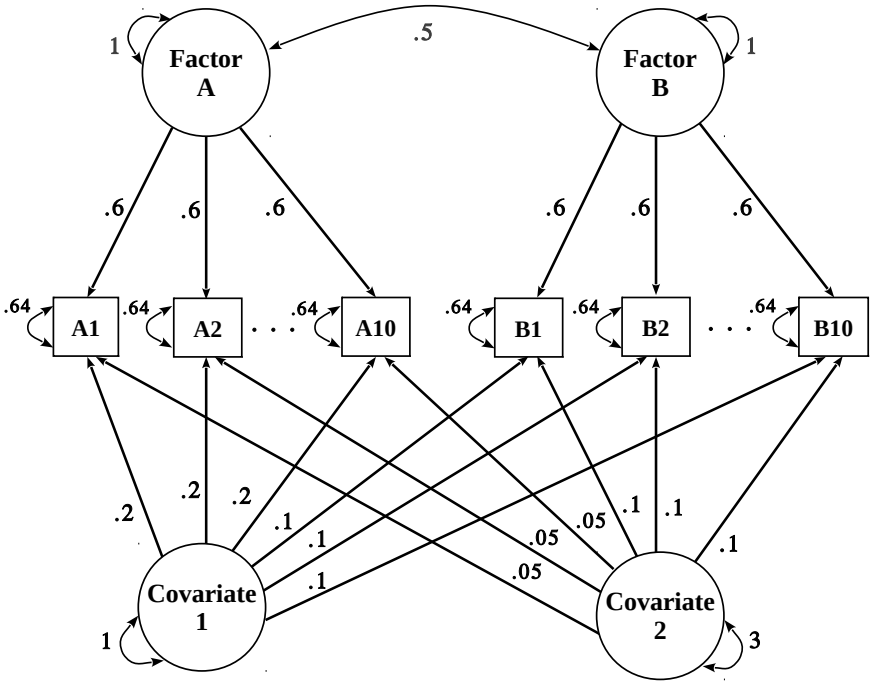

Figure 2.1: Path diagram of the population level data-generation model
Note: Mean Structures are not shown

After the appropriate matrices were populated with the parameter values described above, the latent and residual covariance matrices were included as arguments to the R function rmvtnorm

[2]Note that the mean structures are not pictured in Figure 2.1. Except for the mean of Covariate 2 ($\alpha_{c2} = 4$), all latent means and item intercepts were fixed to zero in the population.

(Genz et al., 2012) to simulate multivariate normal factor scores and error terms, respectively. These factor scores and error terms were then included in the factor analytic data model represented by Equation 2.1 to simulate the final data values for use in subsequent analyses.

$$\mathbf{Y} = \eta \Lambda^T + \Theta \tag{2.1}$$

Where $\mathbf{Y}$ is a $N \times 20$ matrix of simulated data, η is a $N \times 4$ matrix of factor scores, Λ is a 20×4 matrix of factor loadings and Θ is a $N \times 20$ matrix of residual error components. Finally, the factor scores associated with the two covariate factors where appended to $\mathbf{Y}$. Therefore, the final simulated data set $\mathbf{Y}_{complete}$ was a $N \times 22$ matrix consisting of the twenty variables to which subsequent analysis models would be fit and the two auxiliary variables that would be used as predictors of the missing data process.

2.1.1 Parsimony Error

In addition to the parameterization described above, an additional degree of complexity was included in the data generating model. In order to maximize the ecological validity of the current work, an attempt was made to reflect the reality of statistical modeling as a reductionist exercise. To accomplish this, the data were generated according to a model which was more complex than the subsequently specified analysis models, thereby introducing a trivial level of misspecification into every model examined in this study. This "hidden" simplification is analogous to the *parsimony error* introduced by researchers attempting to portray highly complex psychological phenomena with simplified mathematical models. To implement this misspecification, the residual covariance matrix used to create the error components of Equation 2.1 was specified to be Toeplitz in form, rather than the customary diagonal matrix implied by the common factor model. This alteration allowed the introduction of a small residual covariance ($\theta = .03$)[3] between every fifth unique factor.

[3]This value was chosen so that ideal comparison models would demonstrate "very good" model fit without reflecting perfect replication of the simulated data. Specifically, when the *full* model described below is fit to complete data that has been generated in this fashion, $CFI \approx [.97 : .98]$ and $RMSEA \approx [.02 : .04]$.

When these residual covariances were not included in the analysis models, the result was a small degree of model misspecification that was constant across conditions.

2.2 Missing Data Imposition

Missing data were simulated to follow a Missing at Random (MAR) process as defined by Rubin (1976). A function was written in R 2.15[4] which ran iteratively through all twenty of the simulated scale variables. This function first computed the value of the inverse normal cumulative distribution function (CDF) associated with each observation of a predictor of the missingness process (i.e., one of the factor scores for the covariate factors). This provided the quantile on the normal CDF associated with that observation of the covariate. If this value was equal to or less than a threshold chosen to delete the appropriate proportion of observations (i.e., the desired percent missing rescaled to account for the algorithm running twenty times) the associated observation of the currently selected simulated variable was deleted. Because this function imposes the missing data according to this simple probit regression model, the Rubin (1976) definition of a MAR process is replicated as closely as possible. That is, the missingness can be considered a pure random sample of the complete data, after conditioning on the predictor of missingness.

2.3 Comparison Conditions

Three comparison conditions were included against which the performance of the SM technique was judged. As an optimal, control condition, analysis models that were analogous to those fit with the SM technique were fit to the complete data (i.e., data sets which were simulated in the same fashion as those provided to the SM treatment without any missing data imposed). It should be noted that all assessments of performance in the current study are based on the implicit understanding that the optimal solution to a missing data problem will produce results equivalent to those that would have been derived had the data been fully observed. This is an important point to

[4]Code Excerpt 2 shows the source code for the function used to impose the missingness.

clarify, because it essentially rejects the possibility that missing data tools can be used to *improve* the quality of model estimates beyond what would be possible with fully observed data, and, as the reader may realize, this is not always true (e.g., well implemented planned missing data designs can offer higher power and lower parameter bias than their complete-data analogues). However, for the purposes of the current work, the estimated model fit from the complete data conditions was chosen as the ideal against which all missing data treatments would be judged.

Additionally, as an anticipated *upward comparison*, FIML estimation was used to fit the analysis models directly to the incomplete data. FIML was chosen for this purpose because it has been shown to produce estimates which demonstrate optimal statistical properties for a wide range of missing data problems (Enders, 2010; Enders & Bandalos, 2001; Schafer & Graham, 2002). To ensure that the FIML conditions truly represented an optimal comparison, an inclusive modeling strategy was implemented through the application of the saturated correlates approach as described by Graham (2003). To fully satisfy the MAR assumption, the saturated correlates method was used to introduce the factor scores associated with the covariate factors as manifest-level auxiliary variables.

Finally, as an anticipated *downward comparison* the missing data were treated by naively averaging the *m* estimates of model fit to achieve an *ad hoc* assessment of the model's fit to the observed data (i.e., the *Naive* approach). This condition was chosen because it is an intuitive alternative to the SM technique, but it has been shown to perform poorly in previous work (Asparouhov & Muthén, 2010). Because the Naive approach is a solution which could fill the same niche as the SM technique, it was judged to be an option relative to which the performance of the SM technique should be assessed. It is expected that the results derived from the SM technique will fall somewhere in between those derived from the Naive approach and those from the FIML estimation.

2.4 Simulation Parameters

There were two simulation parameters varied in this study: sample size and percent missing. Because initial explorations of the SM technique suggested that employing discretely binned conditions lead to a considerable loss of interesting information, the levels of the current simulation parameters were specified in increments that were deemed small enough to approximate a continuous progression through the range of possible values (i.e., $n = \{100, 120, \ldots, 980, 1000\}$, $pm = \{2, 4, \ldots, 48, 50\}$). Therefore, for every replication there were 1150 crossed levels of sample size and percent missing, this number was deemed adequate to support summary of the results through methods designed for continuous data (e.g., plotting three dimensional response surfaces).

Within each of these 1150 cells, eight separate analysis models were estimated using the R package lavaan (Rosseel, 2012). "Full" and "restricted" CFA models were fit to either the complete data with ordinary maximum likelihood (ML) estimation or to the incomplete data using FIML estimation, the SM technique, or the Naive approach. The full model was a two factor CFA in which each factor was indicated by ten of the twenty simulated indicators, and the latent factors were allowed to freely covary (Figure 2.2 shows the path diagram associated with these models).

The restricted model was identical to the full model except that the latent covariance was fixed to $\psi_{2,1} = 0$. This constraint offered the means to assess Hypothesis 4 by facilitating significance tests of the latent covariance via nested model $\Delta\chi^2$ tests. The analysis models associated with the FIML conditions were identical to those described above except that they also incorporated the predictors of the missing data process via the saturated correlates technique (Figure 2.3 shows the path diagram associated with these models).

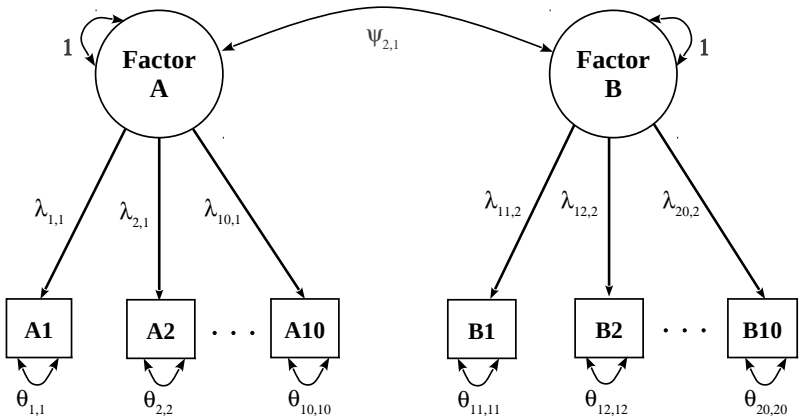

Figure 2.2: Path diagram of the analysis model for imputation-based conditions
Note: Mean Structures are not shown

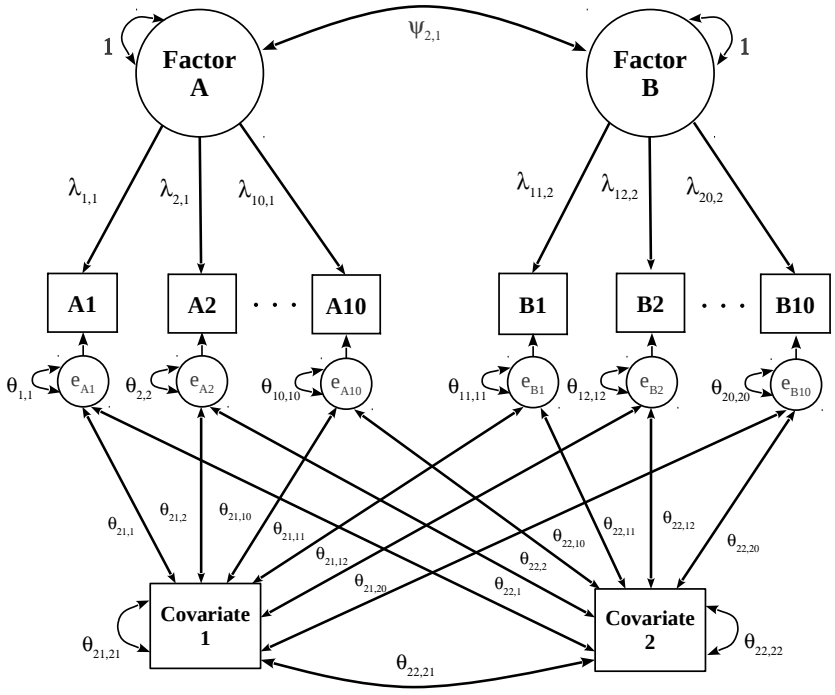

Figure 2.3: Path diagram of the analysis model for FIML-based conditions
Note: Mean Structures are not shown

2.5 Random Numbers

Because of the large number of crossed conditions entailed in this simulation, an effort was made to expedite the analysis process by employing parallel computing to run the simulation code on many processors simultaneously. Consequently, maintaining replicable and non-overlapping random number streams became a paramount concern. Conceptually, if the replications of a Monte Carlo simulation are to be aggregated when summarizing the results, the requirement of non-overlapping random number streams can be thought of as the analogue of the assumption of independent observations in linear regression. To ensure that these stipulations were met, the random numbers were generated via the L'ecuyer RNG (L'ecuyer, Simard, Chen, & Kelton, 2002) as implemented in the R package snowFT (Ševčíková & Rossini, 2010). By maintaining a unique sub-stream for each replication, rather than sending a stream to each processor in the computing cluster, snowFT's implementation of the L'ecuyer RNG ensures that there is no cross over in the random numbers from one replication to another. Thus, each replication can be treated as an independent observation of the population-level process under study.

2.6 Test Statistics

Two test statistics were employed in assessing the performance of the various techniques under study: percentage relative bias (PRB) and root mean square error (RMSE). PRB was chosen because its intuitive interpretation makes it particularly well suited to the *a priori* explication of thresholds by which the adequacy of numeric results can be judged. PRB is simply the average bias in the estimated statistic rescaled as a percentage of the magnitude of the true statistic[5]. For example, $PRB = 10$ would reflect a positive bias in the estimated statistic equivalent to 10 percent

[5]It is important to note that, for this study, the values derived from the complete data conditions were the ideal against which the missing data techniques were judged, so the complete data-based values were considered the "true" values. Thus, the *bias* that is referred to above is not bias in the usual sense because these fit indices are not compared back to any true population values.

of the magnitude of the true statistic. The formula for PRB is quite simple:

$$PRB = 100 \cdot \left(K^{-1} \sum_{i=1}^{K} \frac{\hat{T}_i - T}{T} \right) \tag{2.2}$$

Where T is the true value of the statistic, $\hat{T}_i$ is the estimated statistic for the i^{th} replication and K is the number of replications.

The RMSE was chosen because it combines information on both bias and variability into a well-rounded measure of overall accuracy (Burton, Altman, Royston, & Holder, 2006). Additionally, it is interpretable on the same metric as the statistic under study. The RMSE was therefore considered an ideal statistic to supplement the PRB in this study. The formula for RMSE is also quite simple:

$$RMSE = \sqrt{K^{-1} \sum_{i=1}^{K} \left(\hat{T}_i - T\right)^2} = \sqrt{\left(\bar{\hat{T}} - T\right)^2 + \left(SE_{\hat{T}}\right)^2} \tag{2.3}$$

Where T is the true value of the statistic, $\hat{T}_i$ is the estimated statistic from the i^{th} replication, $\bar{\hat{T}}$ is the mean of the estimated statistic, $SE_{\hat{T}}$ is the empirical standard deviation of the estimated statistic, and K is number of replications.

As a rule-of-thumb by which to judge each techniques performance, $PRB > 5$ was considered to be an excessive degree of bias. In other words, if the estimated value of a missing data-based fit index deviated from the analogous complete data-based value by more than 5% of the magnitude of that complete data-based value, the associated missing data technique was considered to perform unacceptably in that condition.

2.7 Procedure

For every replication a single data set was simulated according to the process described in section 2.1. This data set was then used to fit the complete data control models. Next, the data were treated with the function to impose missingness, as described in section 2.2, thereby producing an incomplete data frame with a certain fixed percent missing. This incomplete data frame was

then used to fit the models for the FIML conditions. Subsequently, the missing data were imputed 100 times using the R package Amelia II (Honaker, King, & Blackwell, 2011). Amelia II is a very powerful software package that employs the Bootstrapped EM algorithm (EMB) to create multiple imputations (Honaker & King, 2010; King, Honaker, Joseph, & Scheve, 2001). These 100 imputed data sets were then either analyzed separately (i.e., using the Naive approach) or submitted to the SM treatment, the results of which were then analyzed. If the Naive approach was used, the resulting fit indices were averaged to get a single, aggregate estimate of model fit.

Once the fit indices of all eight models had been collected, the $\Delta\chi^2$ between the respective full and restricted models was calculated for the complete data conditions and all three missing data treatments, and the PRB and RMSE of the χ^2 , $\Delta\chi^2$, confirmatory fit index (CFI), Tucker-Lewis Index (TLI), root mean error of approximation (RMSEA)[6], and standardized root mean residual (SRMR) were calculated for the SM, FIML, and Naive approaches (i.e., using the complete data values as the *true* values). This process was repeated 500 times for every one of the 1150 crossed levels of percent missing and sample size. This resulted in a total of 3(treatments)$\times$2(model constraints)$\times$46(sample size)$\times$25(percent missing)= 6900 total crossed conditions within each of the 500 replications.

[6]PRB values are not reported for the RMSEA. This is because the formula for PRB is intractable if the value of the true statistic is zero, and this was true of many of the complete data-based RMSEA values.

Chapter 3

Results

3.1 Convergence Rates

Hypothesis 1 was fully supported. All of the imputation-based conditions demonstrated perfect convergence, while the FIML models had very low rates of convergence for conditions with small sample sizes and large percents missing. Figure 3.1 shows the convergence rates of the FIML-based models plotted by sample size and PM, while Table 3.1 shows the convergence rates for the FIML conditions that had lower than 80% convergence.

As the reader can see from the precipitous drop-off in Figure 3.1 and corresponding entries in Table 3.1, sample sizes lower than 200 tend to produce very low convergence rates, particularly when percent missing exceeds 30%.

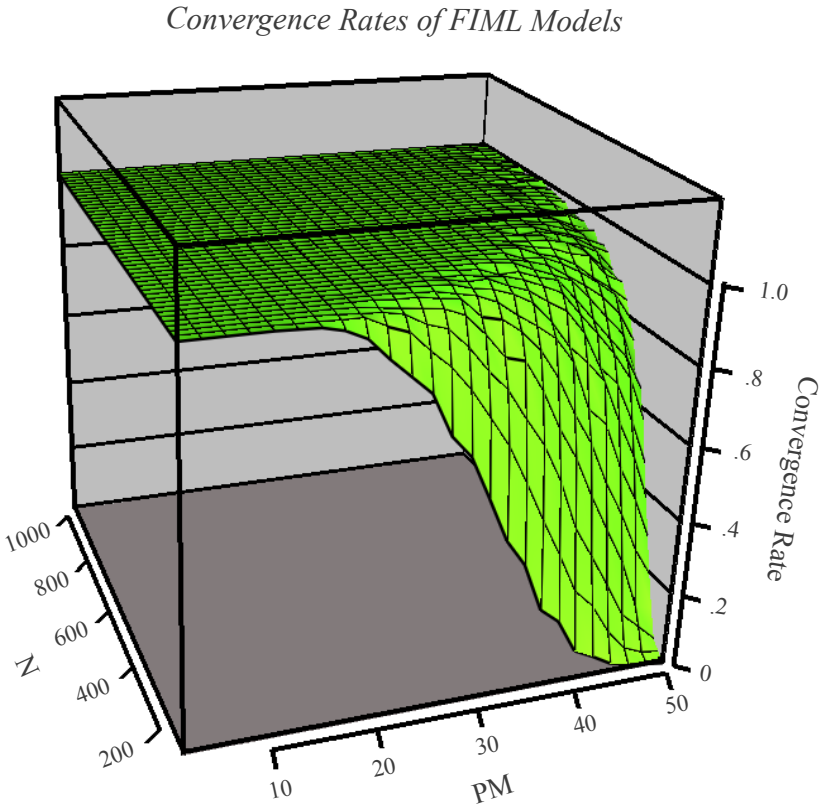

Figure 3.1: Convergence rates for FIML-based models plotted by sample size and percent missing

N	PM	Convergence Rate	N	PM	Convergence Rate	N	PM	Convergence Rate
100	***26***	0.708	***140***	***32***	0.776	***200***	***40***	0.786
100	***28***	0.65	***140***	***34***	0.698	***200***	***42***	0.66
100	***30***	0.538	***140***	***36***	0.626	***200***	***44***	0.614
100	***32***	0.422	***140***	***38***	0.526	***200***	***46***	0.492
100	***34***	0.35	***140***	***40***	0.394	***200***	***48***	0.398
100	***36***	0.22	***140***	***42***	0.308	***200***	***50***	0.282
100	***38***	0.178	***140***	***44***	0.22	***220***	***42***	0.768
100	***40***	0.086	***140***	***46***	0.126	***220***	***44***	0.714
100	***42***	0.06	***140***	***48***	0.082	***220***	***46***	0.602
100	***44***	0.03	***140***	***50***	0.036	***220***	***48***	0.502
100	***46***	0.016	***160***	***36***	0.75	***220***	***50***	0.376
100	***48***	0.008	***160***	***38***	0.648	***240***	***44***	0.738
100	***50***	0	***160***	***40***	0.564	***240***	***46***	0.69
120	***28***	0.8	***160***	***42***	0.456	***240***	***48***	0.586
120	***30***	0.726	***160***	***44***	0.358	***240***	***50***	0.464
120	***32***	0.65	***160***	***46***	0.262	***260***	***46***	0.756
120	***34***	0.55	***160***	***48***	0.13	***260***	***48***	0.686
120	***36***	0.444	***160***	***50***	0.108	***260***	***50***	0.568
120	***38***	0.344	***180***	***38***	0.764	***280***	***46***	0.784
120	***40***	0.236	***180***	***40***	0.666	***280***	***48***	0.73
120	***42***	0.186	***180***	***42***	0.584	***280***	***50***	0.64
120	***44***	0.104	***180***	***44***	0.472	***300***	***48***	0.78
120	***46***	0.046	***180***	***46***	0.38	***300***	***50***	0.654
120	***48***	0.024	***180***	***48***	0.272	***320***	***50***	0.742
120	***50***	0.016	***180***	***50***	0.19	***340***	***50***	0.792

Table 3.1: Convergence rates of FIML conditions with convergence lower than 80%

3.2 Direct Model Fit

Tables 3.2, 3.3, 3.4, 3.5, and 3.6 show the PRB and RMSE values associated with the χ^2, CFI, TLI, SRMR, and RMSEA for selected levels of sample size and percent missing (namely: $n = \{100, 200, \ldots, 1000\}$ and $pm = \{10, 20, \ldots, 50\}$).

Percentage Relative Bias: Chi-Squared										Root Mean Square Error: Chi-Squared									
N	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**
100	***2***	-0.119	2.518	5.322	***600***	***2***	-0.43	1.988	4.799	***100***	***2***	5.715	7.711	12.212	***600***	***2***	5.54	6.96	11.48
100	***10***	0.766	14.419	29.99	***600***	***10***	-2.684	9.655	24.578	***100***	***10***	12.424	30.882	59.369	***600***	***10***	12.971	23.323	52.119
100	***20***	4.301	32.275	68.687	***600***	***20***	-5.011	20.926	53.96	***100***	***20***	18.993	64.812	132.831	***600***	***20***	20.157	47.278	112.956
100	***30***	8.895	48.619	112.75	***600***	***30***	-6.651	33.967	88.746	***100***	***30***	27.048	95.844	216.623	***600***	***30***	24.461	74.12	184.603
100	***40***	22.066	63.457	159.135	***600***	***40***	-8.461	46.611	126.443	***100***	***40***	45.599	123.74	304.874	***600***	***40***	28.77	100.338	262.387
100	***50***	NA	71.336	198.964	***600***	***50***	-8.775	58.311	164.231	***100***	***50***	NA	NA	NA	***600***	***50***	30.931	124.557	340.213
200	***2***	-0.038	2.524	5.444	***700***	***2***	-0.666	1.714	4.465	***200***	***2***	5.34	7.374	11.899	***700***	***2***	5.627	6.676	11.078
200	***10***	-0.213	13.216	29.05	***700***	***10***	-3.298	8.849	23.471	***200***	***10***	12.114	28.115	56.203	***700***	***10***	13.83	22.536	51.21
200	***20***	0.674	29.544	65.33	***700***	***20***	-5.706	20.004	52.328	***200***	***20***	16.935	58.982	124.382	***700***	***20***	20.362	46.358	112.369
200	***30***	2.022	47.247	108.289	***700***	***30***	-7.563	32.638	86.068	***200***	***30***	21.009	92.443	204.947	***700***	***30***	25.611	72.702	183.403
200	***40***	4.392	63.1	153.276	***700***	***40***	-9.179	45.255	122.943	***200***	***40***	23.508	121.378	288.803	***700***	***40***	30.071	100.075	261.768
200	***50***	11.761	75.595	195.214	***700***	***50***	-10.742	55.119	158.288	***200***	***50***	33.517	144.979	367.232	***700***	***50***	33.399	120.966	336.323
300	***2***	0.002	2.529	5.438	***800***	***2***	-0.715	1.706	4.388	***300***	***2***	5.282	7.297	11.902	***800***	***2***	5.518	6.586	11.068
300	***10***	-0.844	12.178	27.879	***800***	***10***	-3.515	8.561	22.818	***300***	***10***	11.685	26.583	55.036	***800***	***10***	13.929	22.501	51.335
300	***20***	-1.713	26.087	61.243	***800***	***20***	-6.535	18.871	50.303	***300***	***20***	16.408	53.588	119.192	***800***	***20***	22.267	45.354	111.235
300	***30***	-1.747	41.921	101.085	***800***	***30***	-8.697	30.743	82.644	***300***	***30***	19.873	84.233	195.723	***800***	***30***	27.86	71.151	181.567
300	***40***	-0.993	57.673	144.493	***800***	***40***	-11.239	42.052	117.482	***300***	***40***	22.278	114.38	278.767	***800***	***40***	33.442	95.829	257.477
300	***50***	1.506	69.946	185.003	***800***	***50***	-12.757	52.344	152.601	***300***	***50***	24.954	137.778	356.012	***800***	***50***	38.285	119.1	334.108
400	***2***	-0.354	2.149	5.044	***900***	***2***	-0.834	1.547	4.15	***400***	***2***	5.335	6.86	11.362	***900***	***2***	5.67	6.535	10.911
400	***10***	-1.712	11.022	26.547	***900***	***10***	-3.779	8.182	22.043	***400***	***10***	12.066	24.766	53.465	***900***	***10***	15.325	22.888	51.299
400	***20***	-3.161	23.805	58.325	***900***	***20***	-6.992	18.106	48.666	***400***	***20***	17.86	50.545	116.044	***900***	***20***	22.662	44.475	110.639
400	***30***	-4.017	38.368	96.128	***900***	***30***	-9.609	29.433	79.856	***400***	***30***	21.504	78.896	189.911	***900***	***30***	29.537	70.023	180.511
400	***40***	-3.439	54.597	138.844	***900***	***40***	-12.128	40.563	113.767	***400***	***40***	24.1	110.885	273.588	***900***	***40***	35.553	95.188	256.488
400	***50***	-3.023	65.712	177.571	***900***	***50***	-14.492	49.101	146.478	***400***	***50***	25.543	132.477	349.15	***900***	***50***	41.321	114.566	329.734
500	***2***	-0.56	1.882	4.747	***1000***	***2***	-0.973	1.392	3.946	***500***	***2***	5.64	6.796	11.19	***1000***	***2***	5.914	6.446	10.737
500	***10***	-2.497	9.912	25.124	***1000***	***10***	-4.121	7.869	21.382	***500***	***10***	12.905	23.546	52.143	***1000***	***10***	15.779	22.532	51.081
500	***20***	-4.219	22.226	55.991	***1000***	***20***	-8.013	16.681	46.455	***500***	***20***	18.742	48.662	114.426	***1000***	***20***	24.963	42.614	108.681
500	***30***	-5.299	36.363	92.452	***1000***	***30***	-11.32	26.871	75.938	***500***	***30***	22.267	76.909	187.684	***1000***	***30***	33.404	66.555	176.705
500	***40***	-5.838	51.067	132.889	***1000***	***40***	-14.303	37.235	108.451	***500***	***40***	24.575	106.428	269.012	***1000***	***40***	40.552	90.878	251.787
500	***50***	-5.921	62.456	170.94	***1000***	***50***	-16.388	46.372	141.055	***500***	***50***	27.158	129.598	345.531	***1000***	***50***	46.57	112.179	326.826

Table 3.2: Percentage Relative Bias and Root Mean Square Error of the χ^2 from selected conditions

When referring to these tables, the reader will see that Hypothesis 2 is only minimally supported. The SM-based absolute fit indices (i.e., χ^2 and RMSEA) and residual-based indices (i.e., SRMR) performed quite poorly under nearly all model parameterizations. However, the SM-based incremental fit indices (i.e., CFI and TLI) performed relatively well, although their performance was degraded in conditions with small sample sizes and large percents missing.

A result of key interest is the support of Hypothesis 3 in the context of assessing direct model

Percentage Relative Bias: CFI										Root Mean Square Error: CFI									
N	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**
100	***2***	0.139	-0.66	-1.434	***600***	***2***	-0.002	-0.111	-0.265	***100***	***2***	0.0114	0.0111	0.0173	***600***	***2***	0.0016	0.0018	0.003
100	***10***	-0.631	-4.014	-8.085	***600***	***10***	0.019	-0.541	-1.356	***100***	***10***	0.0231	0.0444	0.0816	***600***	***10***	0.0034	0.0063	0.0139
100	***20***	-2.705	-8.809	-16.902	***600***	***20***	0.027	-1.192	-2.983	***100***	***20***	0.0434	0.0911	0.1668	***600***	***20***	0.0055	0.013	0.0301
100	***30***	-6.013	-12.987	-25.134	***600***	***30***	-0.008	-1.966	-4.903	***100***	***30***	0.0764	0.1318	0.2463	***600***	***30***	0.0073	0.0207	0.0491
100	***40***	-14.373	-16.781	-32.362	***600***	***40***	-0.035	-2.776	-7.04	***100***	***40***	0.1504	0.1687	0.3161	***600***	***40***	0.0097	0.0288	0.0703
100	***50***	NA	-19.231	-37.752	***600***	***50***	-0.275	-3.636	-9.327	***100***	***50***	NA	NA	NA	***600***	***50***	0.0135	0.0375	0.093
200	***2***	0.072	-0.342	-0.746	***700***	***2***	0.004	-0.086	-0.218	***200***	***2***	0.0056	0.0055	0.0089	***700***	***2***	0.0013	0.0015	0.0025
200	***10***	-0.079	-1.86	-4.095	***700***	***10***	0.033	-0.44	-1.145	***200***	***10***	0.0111	0.0213	0.042	***700***	***10***	0.0029	0.0052	0.0117
200	***20***	-0.537	-4.249	-8.966	***700***	***20***	0.029	-1.014	-2.565	***200***	***20***	0.0178	0.0449	0.09	***700***	***20***	0.0046	0.011	0.0258
200	***30***	-1.243	-6.771	-14.171	***700***	***30***	0.001	-1.677	-4.226	***200***	***30***	0.0267	0.0704	0.1417	***700***	***30***	0.006	0.0176	0.0423
200	***40***	-2.613	-9.148	-19.294	***700***	***40***	-0.066	-2.401	-6.107	***200***	***40***	0.0391	0.0934	0.192	***700***	***40***	0.0084	0.0249	0.061
200	***50***	-6.062	-11.154	-23.774	***700***	***50***	-0.156	-3.07	-8.06	***200***	***50***	0.0729	0.1138	0.2363	***700***	***50***	0.011	0.0317	0.0804
300	***2***	0.006	-0.242	-0.529	***800***	***2***	0.002	-0.077	-0.194	***300***	***2***	0.0034	0.0037	0.0061	***800***	***2***	0.0011	0.0013	0.0022
300	***10***	-0.049	-1.218	-2.773	***800***	***10***	0.021	-0.389	-1.01	***300***	***10***	0.007	0.0139	0.0283	***800***	***10***	0.0026	0.0046	0.0103
300	***20***	-0.14	-2.638	-5.983	***800***	***20***	0.042	-0.866	-2.232	***300***	***20***	0.0108	0.0281	0.0601	***800***	***20***	0.0041	0.0095	0.0225
300	***30***	-0.403	-4.286	-9.662	***800***	***30***	0.017	-1.439	-3.692	***300***	***30***	0.0158	0.0446	0.0965	***800***	***30***	0.0055	0.0152	0.0369
300	***40***	-0.924	-5.985	-13.542	***800***	***40***	0.023	-2.038	-5.333	***300***	***40***	0.0221	0.0615	0.1349	***800***	***40***	0.007	0.0212	0.0533
300	***50***	-2.077	-7.46	-17.175	***800***	***50***	-0.07	-2.663	-7.116	***300***	***50***	0.0342	0.0764	0.171	***800***	***50***	0.0102	0.0276	0.0711
400	***2***	0.024	-0.162	-0.381	***900***	***2***	0.003	-0.065	-0.169	***400***	***2***	0.0025	0.0027	0.0044	***900***	***2***	0.001	0.0012	0.002
400	***10***	0.014	-0.853	-2.045	***900***	***10***	0.018	-0.342	-0.896	***400***	***10***	0.0051	0.0098	0.0209	***900***	***10***	0.0025	0.0042	0.0092
400	***20***	-0.015	-1.875	-4.464	***900***	***20***	0.031	-0.767	-1.987	***400***	***20***	0.0083	0.0202	0.0449	***900***	***20***	0.0034	0.0083	0.02
400	***30***	-0.118	-3.066	-7.284	***900***	***30***	0.019	-1.272	-3.289	***400***	***30***	0.011	0.0321	0.0729	***900***	***30***	0.0047	0.0134	0.0329
400	***40***	-0.478	-4.438	-10.42	***900***	***40***	0.007	-1.813	-4.767	***400***	***40***	0.0156	0.0459	0.104	***900***	***40***	0.0063	0.0188	0.0476
400	***50***	-1.003	-5.555	-13.405	***900***	***50***	-0.016	-2.314	-6.33	***400***	***50***	0.0218	0.0571	0.1337	***900***	***50***	0.0082	0.0239	0.0631
500	***2***	0.015	-0.119	-0.301	***1000***	***2***	0.005	-0.054	-0.149	***500***	***2***	0.0019	0.0021	0.0035	***1000***	***2***	0.0009	0.001	0.0017
500	***10***	0.035	-0.639	-1.607	***1000***	***10***	0.018	-0.306	-0.806	***500***	***10***	0.0041	0.0075	0.0165	***1000***	***10***	0.0022	0.0037	0.0083
500	***20***	0.009	-1.471	-3.592	***1000***	***20***	0.052	-0.66	-1.765	***500***	***20***	0.0066	0.0159	0.0362	***1000***	***20***	0.0032	0.0072	0.0178
500	***30***	-0.072	-2.433	-5.886	***1000***	***30***	0.074	-1.086	-2.915	***500***	***30***	0.0086	0.0254	0.0589	***1000***	***30***	0.0045	0.0115	0.0292
500	***40***	-0.258	-3.509	-8.483	***1000***	***40***	0.086	-1.565	-4.254	***500***	***40***	0.0113	0.0361	0.0846	***1000***	***40***	0.0061	0.0164	0.0425
500	***50***	-0.578	-4.46	-11.044	***1000***	***50***	0.049	-2.054	-5.718	***500***	***50***	0.0163	0.0457	0.1101	***1000***	***50***	0.008	0.0213	0.057

Table 3.3: Percentage Relative Bias and Root Mean Square Error of the CFI from selected conditions

Percentage Relative Bias: TLI										Root Mean Square Error: TLI									
N	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**
100	***2***	-0.066	-0.835	-1.762	***600***	***2***	-0.008	-0.126	-0.3	***100***	***2***	0.0101	0.0133	0.0208	***600***	***2***	0.0017	0.0021	0.0034
100	***10***	-0.935	-4.739	-9.337	***600***	***10***	0.016	-0.613	-1.532	***100***	***10***	0.0256	0.0518	0.0939	***600***	***10***	0.0038	0.0071	0.0156
100	***20***	-3.291	-10.149	-19.273	***600***	***20***	0.025	-1.347	-3.363	***100***	***20***	0.05	0.1048	0.1901	***600***	***20***	0.0062	0.0147	0.0339
100	***30***	-7.039	-14.857	-28.553	***600***	***30***	-0.015	-2.218	-5.525	***100***	***30***	0.0882	0.1509	0.2795	***600***	***30***	0.0082	0.0233	0.0553
100	***40***	-16.39	-19.135	-36.702	***600***	***40***	-0.045	-3.13	-7.93	***100***	***40***	0.1704	0.1922	0.3579	***600***	***40***	0.0109	0.0324	0.0791
100	***50***	NA	-21.892	-42.779	***600***	***50***	-0.314	-4.098	-10.504	***100***	***50***	NA	NA	NA	***600***	***50***	0.0151	0.0422	0.1047
200	***2***	-0.051	-0.432	-0.921	***700***	***2***	0.002	-0.096	-0.246	***200***	***2***	0.0049	0.0067	0.0106	***700***	***2***	0.0014	0.0017	0.0028
200	***10***	-0.219	-2.215	-4.739	***700***	***10***	0.035	-0.497	-1.29	***200***	***10***	0.0123	0.025	0.0484	***700***	***10***	0.0033	0.0059	0.0132
200	***20***	-0.735	-4.912	-10.22	***700***	***20***	0.031	-1.144	-2.889	***200***	***20***	0.0203	0.0518	0.1025	***700***	***20***	0.0051	0.0124	0.029
200	***30***	-1.533	-7.75	-16.078	***700***	***30***	-0.001	-1.889	-4.759	***200***	***30***	0.0305	0.0804	0.1606	***700***	***30***	0.0068	0.0198	0.0475
200	***40***	-3.073	-10.424	-21.842	***700***	***40***	-0.076	-2.704	-6.877	***200***	***40***	0.0448	0.1063	0.2172	***700***	***40***	0.0095	0.0281	0.0686
200	***50***	-6.956	-12.683	-26.884	***700***	***50***	-0.177	-3.458	-9.075	***200***	***50***	0.0829	0.1293	0.267	***700***	***50***	0.0124	0.0357	0.0905
300	***2***	-0.043	-0.294	-0.629	***800***	***2***	0.001	-0.086	-0.218	***300***	***2***	0.0032	0.0044	0.0071	***800***	***2***	0.0012	0.0015	0.0025
300	***10***	-0.105	-1.418	-3.171	***800***	***10***	0.022	-0.44	-1.139	***300***	***10***	0.0077	0.016	0.0323	***800***	***10***	0.0029	0.0052	0.0116
300	***20***	-0.207	-3.019	-6.784	***800***	***20***	0.046	-0.976	-2.514	***300***	***20***	0.0121	0.032	0.0681	***800***	***20***	0.0046	0.0107	0.0253
300	***30***	-0.504	-4.874	-10.926	***800***	***30***	0.018	-1.621	-4.158	***300***	***30***	0.0177	0.0505	0.109	***800***	***30***	0.0062	0.0171	0.0415
300	***40***	-1.091	-6.787	-15.293	***800***	***40***	0.024	-2.295	-6.004	***300***	***40***	0.025	0.0696	0.1522	***800***	***40***	0.0079	0.0238	0.0599
300	***50***	-2.386	-8.447	-19.382	***800***	***50***	-0.08	-2.999	-8.012	***300***	***50***	0.0387	0.0864	0.1928	***800***	***50***	0.0114	0.0311	0.0799
400	***2***	-0.006	-0.193	-0.448	***900***	***2***	0.003	-0.073	-0.19	***400***	***2***	0.0024	0.0031	0.0051	***900***	***2***	0.0011	0.0013	0.0022
400	***10***	-0.017	-0.988	-2.334	***900***	***10***	0.02	-0.385	-1.008	***400***	***10***	0.0056	0.0113	0.0238	***900***	***10***	0.0028	0.0047	0.0104
400	***20***	-0.049	-2.143	-5.058	***900***	***20***	0.034	-0.863	-2.237	***400***	***20***	0.0093	0.023	0.0508	***900***	***20***	0.0039	0.0093	0.0225
400	***30***	-0.166	-3.484	-8.231	***900***	***30***	0.022	-1.432	-3.702	***400***	***30***	0.0123	0.0364	0.0823	***900***	***30***	0.0053	0.015	0.037
400	***40***	-0.569	-5.027	-11.761	***900***	***40***	0.007	-2.041	-5.366	***400***	***40***	0.0176	0.0519	0.1173	***900***	***40***	0.0071	0.0212	0.0535
400	***50***	-1.158	-6.284	-15.121	***900***	***50***	-0.019	-2.605	-7.125	***400***	***50***	0.0245	0.0645	0.1507	***900***	***50***	0.0092	0.0269	0.071
500	***2***	0.005	-0.138	-0.346	***1000***	***2***	0.006	-0.061	-0.168	***500***	***2***	0.002	0.0024	0.004	***1000***	***2***	0.001	0.0012	0.0019
500	***10***	0.027	-0.731	-1.822	***1000***	***10***	0.021	-0.344	-0.908	***500***	***10***	0.0046	0.0086	0.0186	***1000***	***10***	0.0025	0.0042	0.0093
500	***20***	-0.001	-1.668	-4.055	***1000***	***20***	0.059	-0.743	-1.987	***500***	***20***	0.0074	0.018	0.0408	***1000***	***20***	0.0036	0.0081	0.02
500	***30***	-0.093	-2.751	-6.638	***1000***	***30***	0.083	-1.222	-3.281	***500***	***30***	0.0097	0.0287	0.0663	***1000***	***30***	0.005	0.013	0.0328
500	***40***	-0.302	-3.962	-9.561	***1000***	***40***	0.097	-1.761	-4.789	***500***	***40***	0.0128	0.0407	0.0952	***1000***	***40***	0.0069	0.0184	0.0478
500	***50***	-0.663	-5.032	-12.445	***1000***	***50***	0.055	-2.313	-6.436	***500***	***50***	0.0183	0.0516	0.1239	***1000***	***50***	0.009	0.024	0.0641

Table 3.4: Percentage Relative Bias and Root Mean Square Error of the TLI from selected conditions

Percentage Relative Bias: SRMR										Root Mean Square Error: SRMR									
N	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**
100	***2***	-6.747	1.287	2.617	***600***	***2***	-10.527	1.095	2.479	***100***	***2***	0.0049	0.0016	0.0023	***600***	***2***	0.0031	0.0006	0.0009
100	***10***	3.749	7.092	13.722	***600***	***10***	-4.516	5.467	12.311	***100***	***10***	0.0043	0.0056	0.0096	***600***	***10***	0.0017	0.0019	0.0037
100	***20***	21.639	15.293	29.056	***600***	***20***	5.027	11.71	25.701	***100***	***20***	0.0154	0.0111	0.0198	***600***	***20***	0.0022	0.0037	0.0075
100	***30***	47.005	23.792	45.827	***600***	***30***	18.335	19.014	40.54	***100***	***30***	0.0322	0.0168	0.0309	***600***	***30***	0.0056	0.0058	0.0117
100	***40***	85.914	33.741	64.807	***600***	***40***	37.729	27.004	56.551	***100***	***40***	0.0583	0.0234	0.0436	***600***	***40***	0.0111	0.0081	0.0163
100	***50***	NA	42.826	84.671	***600***	***50***	69.217	35.334	73.171	***100***	***50***	NA	NA	NA	***600***	***50***	0.0202	0.0105	0.0211
200	***2***	-9.22	1.286	2.676	***700***	***2***	-10.705	0.976	2.344	***200***	***2***	0.0045	0.0011	0.0016	***700***	***2***	0.0029	0.0005	0.0008
200	***10***	-1.241	6.664	13.547	***700***	***10***	-4.966	5.156	11.968	***200***	***10***	0.0022	0.0038	0.0067	***700***	***10***	0.0016	0.0017	0.0033
200	***20***	12.085	14.492	28.474	***700***	***20***	4.666	11.556	25.429	***200***	***20***	0.0067	0.0076	0.0139	***700***	***20***	0.0018	0.0034	0.0069
200	***30***	30.976	23.011	44.783	***700***	***30***	17.336	18.427	39.778	***200***	***30***	0.0154	0.0116	0.0216	***700***	***30***	0.005	0.0052	0.0107
200	***40***	58.793	32.182	62.337	***700***	***40***	36.122	26.323	55.599	***200***	***40***	0.0286	0.016	0.0299	***700***	***40***	0.0099	0.0074	0.015
200	***50***	108.014	41.78	80.84	***700***	***50***	65.343	33.926	71.59	***200***	***50***	0.0522	0.0206	0.0388	***700***	***50***	0.0178	0.0094	0.0192
300	***2***	-9.832	1.258	2.651	***800***	***2***	-10.826	0.916	2.265	***300***	***2***	0.004	0.0009	0.0013	***800***	***2***	0.0028	0.0005	0.0007
300	***10***	-2.705	6.314	13.244	***800***	***10***	-5.027	5.168	11.868	***300***	***10***	0.0019	0.003	0.0054	***800***	***10***	0.0015	0.0016	0.0031
300	***20***	8.484	13.184	27.34	***800***	***20***	3.795	10.875	24.583	***300***	***20***	0.0041	0.0057	0.011	***800***	***20***	0.0016	0.003	0.0063
300	***30***	24.219	20.915	42.81	***800***	***30***	16.394	17.898	38.981	***300***	***30***	0.01	0.0087	0.017	***800***	***30***	0.0045	0.0048	0.0099
300	***40***	47.976	29.619	59.796	***800***	***40***	34.488	25.299	54.308	***300***	***40***	0.0194	0.0122	0.0237	***800***	***40***	0.009	0.0067	0.0138
300	***50***	88.335	38.894	77.566	***800***	***50***	62.798	33.014	70.315	***300***	***50***	0.0352	0.0159	0.0307	***800***	***50***	0.0162	0.0087	0.0179
400	***2***	-10.26	1.088	2.49	***900***	***2***	-10.842	0.96	2.276	***400***	***2***	0.0036	0.0007	0.001	***900***	***2***	0.0027	0.0004	0.0007
400	***10***	-3.642	5.931	12.902	***900***	***10***	-5.245	5.029	11.604	***400***	***10***	0.0018	0.0024	0.0046	***900***	***10***	0.0015	0.0015	0.0029
400	***20***	6.765	12.412	26.62	***900***	***20***	3.532	10.748	24.222	***400***	***20***	0.0031	0.0047	0.0093	***900***	***20***	0.0014	0.0028	0.0059
400	***30***	21.71	20.195	42.144	***900***	***30***	15.803	17.425	38.206	***400***	***30***	0.0079	0.0074	0.0146	***900***	***30***	0.0041	0.0045	0.0093
400	***40***	43.939	29.131	59.099	***900***	***40***	33.281	24.869	53.37	***400***	***40***	0.0155	0.0105	0.0205	***900***	***40***	0.0083	0.0063	0.013
400	***50***	79.458	37.105	75.74	***900***	***50***	60.005	32.059	68.845	***400***	***50***	0.0278	0.0132	0.0262	***900***	***50***	0.0147	0.008	0.0167
500	***2***	-10.452	1.055	2.457	***1000***	***2***	-10.954	0.876	2.175	***500***	***2***	0.0033	0.0006	0.0009	***1000***	***2***	0.0026	0.0004	0.0006
500	***10***	-4.165	5.611	12.53	***1000***	***10***	-5.334	5.025	11.473	***500***	***10***	0.0018	0.0021	0.004	***1000***	***10***	0.0015	0.0014	0.0028
500	***20***	5.87	12.131	26.266	***1000***	***20***	3.024	10.306	23.583	***500***	***20***	0.0024	0.0041	0.0083	***1000***	***20***	0.0013	0.0026	0.0055
500	***30***	19.84	19.618	41.312	***1000***	***30***	14.235	16.247	36.803	***500***	***30***	0.0065	0.0064	0.013	***1000***	***30***	0.0036	0.004	0.0086
500	***40***	41.134	28.492	58.165	***1000***	***40***	31.769	23.702	51.896	***500***	***40***	0.0131	0.0092	0.0182	***1000***	***40***	0.0076	0.0058	0.0121
500	***50***	73.398	36.242	74.392	***1000***	***50***	58.162	31.156	67.434	***500***	***50***	0.0232	0.0117	0.0232	***1000***	***50***	0.0137	0.0075	0.0157

Table 3.5: Percentage Relative Bias and Root Mean Square Error of the SRMR from selected conditions

N	PM	FIML	SM	Naive	N	PM	FIML	SM	Naive
100	***2***	0.0056	0.0072	0.0104	***600***	***2***	0.0017	0.0021	0.0032
100	***10***	0.0118	0.0223	0.0362	***600***	***10***	0.0039	0.0059	0.0113
100	***20***	0.0164	0.0387	0.0628	***600***	***20***	0.0065	0.0105	0.0201
100	***30***	0.0214	0.0507	0.0858	***600***	***30***	0.008	0.0148	0.0283
100	***40***	0.0317	0.0601	0.1059	***600***	***40***	0.0092	0.0184	0.0358
100	***50***	NA	NA	NA	***600***	***50***	0.0102	0.0215	0.0425
200	***2***	0.0039	0.005	0.0075	***700***	***2***	0.0015	0.0018	0.0026
200	***10***	0.0081	0.0154	0.0257	***700***	***10***	0.0036	0.0049	0.0097
200	***20***	0.0112	0.0265	0.0437	***700***	***20***	0.0056	0.0089	0.0177
200	***30***	0.0134	0.036	0.0599	***700***	***30***	0.0072	0.0128	0.0252
200	***40***	0.0148	0.0431	0.0739	***700***	***40***	0.0087	0.0162	0.0322
200	***50***	0.019	0.0483	0.0853	***700***	***50***	0.0097	0.0187	0.038
300	***2***	0.0028	0.0038	0.0058	***800***	***2***	0.0012	0.0014	0.0022
300	***10***	0.0064	0.0113	0.0196	***800***	***10***	0.0032	0.0042	0.0085
300	***20***	0.0088	0.0193	0.0334	***800***	***20***	0.0052	0.0076	0.0157
300	***30***	0.0103	0.0263	0.046	***800***	***30***	0.0069	0.011	0.0225
300	***40***	0.0112	0.0325	0.0575	***800***	***40***	0.0086	0.014	0.0288
300	***50***	0.0125	0.0368	0.0668	***800***	***50***	0.0099	0.0165	0.0344
400	***2***	0.0026	0.0031	0.0046	***900***	***2***	0.0011	0.0012	0.0019
400	***10***	0.005	0.0087	0.0158	***900***	***10***	0.0031	0.0037	0.0076
400	***20***	0.0079	0.015	0.0273	***900***	***20***	0.0047	0.0067	0.0142
400	***30***	0.0096	0.0209	0.0379	***900***	***30***	0.0064	0.0098	0.0205
400	***40***	0.0103	0.0265	0.0479	***900***	***40***	0.008	0.0126	0.0264
400	***50***	0.011	0.0299	0.0557	***900***	***50***	0.0096	0.0145	0.0314
500	***2***	0.0023	0.0025	0.0038	***1000***	***2***	0.001	0.001	0.0016
500	***10***	0.0046	0.007	0.013	***1000***	***10***	0.0027	0.0033	0.0068
500	***20***	0.0069	0.0123	0.0231	***1000***	***20***	0.0047	0.0059	0.0128
500	***30***	0.0083	0.0174	0.0324	***1000***	***30***	0.0065	0.0086	0.0186
500	***40***	0.0092	0.0219	0.041	***1000***	***40***	0.0085	0.011	0.024
500	***50***	0.01	0.0252	0.0481	***1000***	***50***	0.0098	0.013	0.0289

Table 3.6: Root Mean Square Error of the RMSEA from selected conditions

fit. The Naive approach performed universally more poorly than the SM technique in all conditions tested. To give a visualization of the relative performance of the SM and Naive approaches vis-à-vis the complete data conditions, Figure 3.2 shows superimposed surfaces that correspond to the CFI and TLI of these three conditions plotted by sample size and PM. Referring to this figure, the reader can see a representative example of the consistent superiority of the SM technique to the Naive approach.

Plate 1: CFI for the SuperMatrix, Naive, and Complete Data Conditions

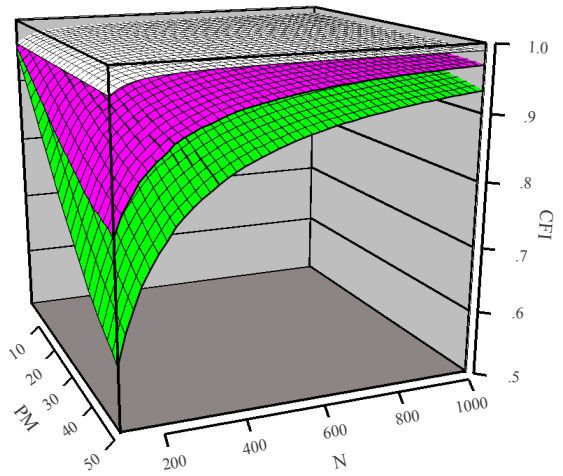

Plate 2: TLI for the SuperMatrix, Naive, and Complete Data Conditions

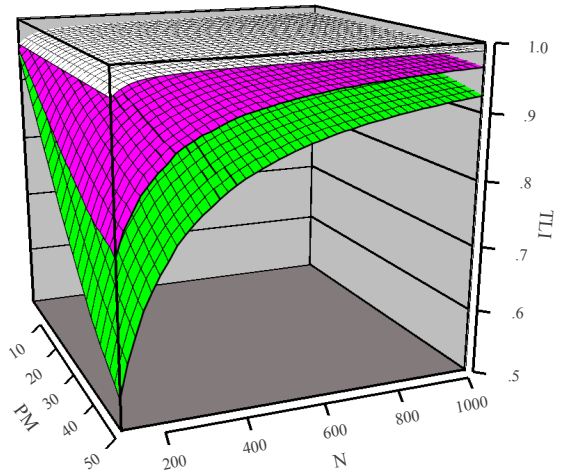

Figure 3.2: Incremental fit indices for complete data, SM, & naive conditions plotted by sample size and percent missing
Upper surface=Complete data conditions, Intermediate surface=SM conditions, Lower surface=Naive conditions

3.3 Significance Testing with the $\Delta\chi^2$

Perhaps the most important finding of this study is the complete support of Hypothesis 4. As the reader can see from Table 3.7, the SM-based $\Delta\chi^2$ values were reasonably accurate across all conditions tested (although these values tend to demonstrate an unacceptable degree of positive bias when $pm > 40$ and $n < 300$). While the Naive-based $\Delta\chi^2$ also acquits itself reasonably well, a direct comparison between the PRB and RMSE values associated with the SM and Naive conditions shows that the SM technique leads to consistently superior results. Therefore, Hypothesis 3 is also supported in the context of significance testing with the $\Delta\chi^2$ test. An unexpected outcome was the poor performance of the FIML-based $\Delta\chi^2$ tests. As the reader can see from Table 3.7 the FIML-based $\Delta\chi^2$ values showed a consistent and appreciable negative bias. In fact, for all levels of sample size, the FIML-based $\Delta\chi^2$ values showed an unacceptable degree of negative bias when $pm > 10$. Plate 1 of Figure 3.3 shows how closely, on average, the SM-based $\Delta\chi^2$ values track their complete data-based counterparts. On the other hand, Plate 2 of Figure 3.3 illustrates the considerable discrepancy between the average FIML-based $\Delta\chi^2$ values and the complete data-based versions.

Percentage Relative Bias: Chi-Squared Difference										Root Mean Square Error: Chi-Squared Difference									
N	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**	**N**	**PM**	**FIML**	**SM**	**Naive**
100	***2***	-1.201	0.301	0.368	***600***	***2***	-1.338	0.016	0.029	***100***	***2***	1.255	1.238	1.242	***600***	***2***	3.443	2.76	2.76
100	***10***	-5.941	1.903	2.286	***600***	***10***	-6.023	0.8	0.86	***100***	***10***	3.215	3.007	3.021	***600***	***10***	10.646	6.112	6.13
100	***20***	-13.876	2.924	3.792	***600***	***20***	-13.302	1.137	1.269	***100***	***20***	5.391	4.644	4.693	***600***	***20***	22.074	9.838	9.872
100	***30***	-23.867	3.892	5.485	***600***	***30***	-21.173	2.104	2.357	***100***	***30***	7.647	5.704	5.815	***600***	***30***	33.942	13.556	13.649
100	***40***	-33.526	6.527	9.493	***600***	***40***	-30.383	2.946	3.361	***100***	***40***	9.948	7.942	8.216	***600***	***40***	47.345	17.353	17.545
100	***50***	NA	14.522	20.885	***600***	***50***	-40.177	5.389	6.168	***100***	***50***	NA	10.693	11.483	***600***	***50***	62.45	22.398	22.86
200	***2***	-1.405	0.066	0.102	***700***	***2***	-1.269	0.064	0.074	***200***	***2***	1.839	1.661	1.659	***700***	***2***	3.633	2.935	2.935
200	***10***	-6.216	1.081	1.259	***700***	***10***	-6.465	0.414	0.465	***200***	***10***	4.951	4.016	4.025	***700***	***10***	12.912	6.674	6.687
200	***20***	-13.204	2.169	2.582	***700***	***20***	-13.212	1.301	1.414	***200***	***20***	8.882	6.351	6.398	***700***	***20***	25.027	10.969	11.015
200	***30***	-21.675	3.2	3.934	***700***	***30***	-20.918	2.396	2.605	***200***	***30***	13.146	8.514	8.601	***700***	***30***	38.555	14.984	15.097
200	***40***	-30.463	6.132	7.465	***700***	***40***	-30.205	2.895	3.254	***200***	***40***	16.955	10.199	10.447	***700***	***40***	54.953	18.113	18.299
200	***50***	-40.791	8.764	11.178	***700***	***50***	-39.898	5.628	6.284	***200***	***50***	22.668	13.622	14.133	***700***	***50***	71.725	25.415	25.942
300	***2***	-1.295	0.085	0.108	***800***	***2***	-1.226	0.105	0.114	***300***	***2***	2.15	1.945	1.944	***800***	***2***	3.932	3.061	3.061
300	***10***	-6.52	0.519	0.636	***800***	***10***	-6.193	0.549	0.593	***300***	***10***	6.765	4.728	4.735	***800***	***10***	14.22	7.225	7.242
300	***20***	-13.157	1.652	1.913	***800***	***20***	-13.231	1.353	1.452	***300***	***20***	12.094	7.561	7.59	***800***	***20***	28.421	11.895	11.95
300	***30***	-21.594	2.361	2.831	***800***	***30***	-21.177	2.066	2.245	***300***	***30***	18.491	9.441	9.509	***800***	***30***	44.801	16.221	16.308
300	***40***	-31.343	3.659	4.499	***800***	***40***	-30.273	2.481	2.794	***300***	***40***	25.444	11.814	12.018	***800***	***40***	62.487	19.608	19.799
300	***50***	-40.537	7.761	9.321	***800***	***50***	-40.482	4.327	4.88	***300***	***50***	32.59	16.281	16.807	***800***	***50***	82.82	25.716	26.146
400	***2***	-1.19	0.183	0.2	***900***	***2***	-1.157	0.182	0.19	***400***	***2***	2.574	2.297	2.297	***900***	***2***	4.242	3.301	3.303
400	***10***	-6.271	0.692	0.78	***900***	***10***	-6.235	0.595	0.635	***400***	***10***	8.124	5.208	5.218	***900***	***10***	16.165	8.237	8.253
400	***20***	-13.07	1.635	1.833	***900***	***20***	-13.336	1.053	1.142	***400***	***20***	15.211	8.112	8.153	***900***	***20***	31.93	11.815	11.852
400	***30***	-21.536	2.26	2.622	***900***	***30***	-21.298	1.659	1.818	***400***	***30***	23.485	10.365	10.452	***900***	***30***	50.013	15.729	15.819
400	***40***	-30.841	3.237	3.875	***900***	***40***	-29.99	3.099	3.377	***400***	***40***	32.87	14.63	14.817	***900***	***40***	69.544	21.265	21.489
400	***50***	-40.478	5.714	6.824	***900***	***50***	-39.948	4.599	5.1	***400***	***50***	42.645	17.094	17.529	***900***	***50***	91.792	27.673	28.133
500	***2***	-1.186	0.18	0.194	***1000***	***2***	-1.261	0.064	0.071	***500***	***2***	2.866	2.54	2.542	***1000***	***2***	4.729	3.576	3.577
500	***10***	-6.3	0.626	0.698	***1000***	***10***	-6.197	0.612	0.647	***500***	***10***	9.725	5.878	5.89	***1000***	***10***	17.559	8.48	8.496
500	***20***	-13.991	0.661	0.824	***1000***	***20***	-13.679	0.616	0.697	***500***	***20***	19.369	8.943	8.966	***1000***	***20***	36.195	12.873	12.901
500	***30***	-21.317	2.082	2.376	***1000***	***30***	-21.195	1.673	1.817	***500***	***30***	28.819	12.501	12.58	***1000***	***30***	55.106	17.365	17.452
500	***40***	-30.836	2.836	3.333	***1000***	***40***	-30.535	2.212	2.468	***500***	***40***	40.489	15.061	15.245	***1000***	***40***	78.533	21.547	21.725
500	***50***	-40.296	5.327	6.23	***1000***	***50***	-40.007	4.645	5.086	***500***	***50***	52.097	19.127	19.558	***1000***	***50***	101.77	29.755	30.242

Table 3.7: Percentage Relative Bias and Root Mean Square Error of the $\Delta\chi^2$ from selected conditions

Plate 1: $\Delta\chi^2$ for the Complete Data and SuperMatrix Conditions

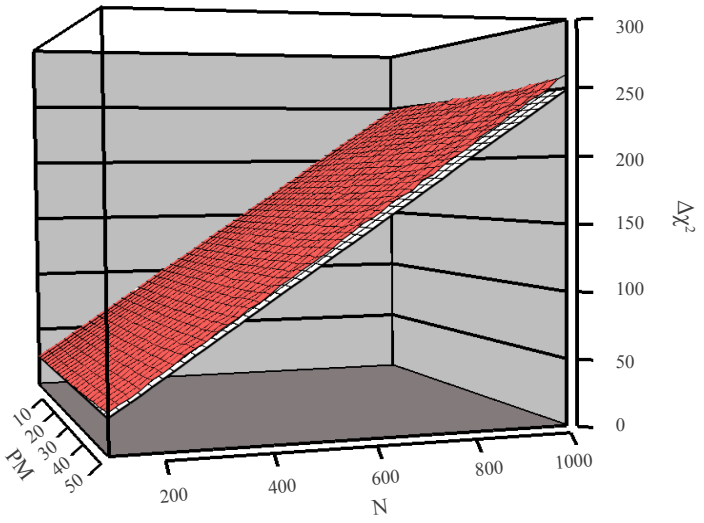

Plate 2: $\Delta\chi^2$ for the Complete Data and FIML Conditions

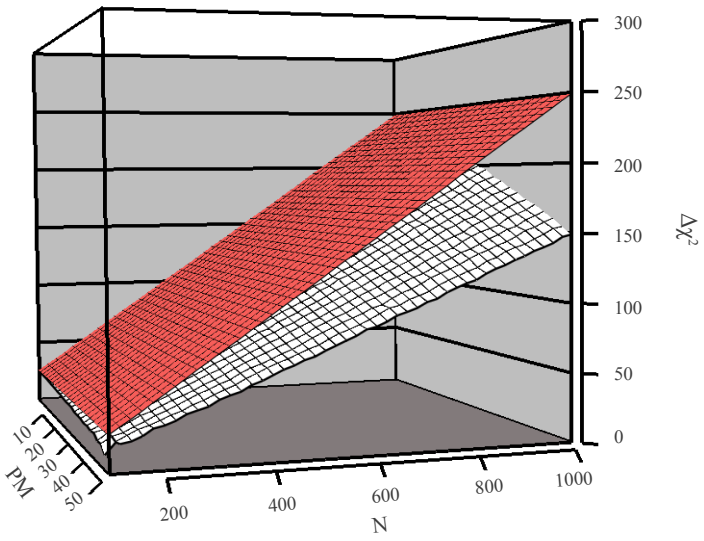

Figure 3.3: $\Delta\chi^2$ values for complete data, SM, & FIML conditions
Plate 1: Upper surface=SM conditions, Lower surface=Complete data conditions
Plate 2: Upper surface=Complete data conditions, Lower surface=FIML conditions

Chapter 4

Discussion

This study endeavored to demonstrate the utility of the SM technique as a parsimonious method of streamlining missing data analysis for the typical social scientist. Of principle concern was the degree to which model fit indices derived from the SM technique could replicate fit indices derived from complete data and the extent to which hypothesis tests conducted with SM-based $\Delta\chi^2$ statistics replicated those conducted with complete data-based $\Delta\chi^2$ statistics. By way of comparison, the performance of the SM technique was contrasted with the performance of FIML estimation and the Naive approach in replicating these same complete data values.

Overall, the SM technique performed reasonably well. As anticipated, the SM technique universally out-performed the Naive approach as both a method of assessing the analysis models' direct fit to the data and as a tool with which to conduct $\Delta\chi^2$-based hypothesis testing. Although many of the SM-based and Naive-based model fit estimates were similar, the SM-based estimates were always superior (i.e., closer to the complete data values). In addition to this superior performance, the SM technique only requires the fitting of a single analysis model (as opposed to m analysis models), so it is more computationally efficient than the Naive approach. The SM technique also offers a more parsimonious implementation with fewer chances for miscalculation than the Naive approach. Thus, there is no reason to consider the Naive approach as a viable alternative to the SM technique.

Unfortunately, though the SM technique performed very well when contrasted with the Naive approach, when compared to the current *gold standard* in missing data analysis, FIML estimation, the results were mixed. On the positive side, the SM technique exhibited very good rates of convergence. In fact, all SM-based conditions achieved 100% convergence. FIML conditions, however, had considerable issues with convergence, particularly under suboptimal model parameterizations. Once sample sizes dropped below 200 and rates of missing increased much beyond 30%, the FIML models began to exhibit very low convergence rates. This is not a terribly surprising results when one considers how FIML estimation is accomplished. Since FIML estimates are derived from aggregated casewise loglikelihoods that are based on only the observed responses, mixing such high percents missing with small sample sizes effectively decreases the sample size below what is required to satisfy the large sample assumption of ML estimation. Because imputation techniques (including the SM technique) operate by simulating plausible values for the missing data, they circumvent this loss of sample size. However, the merits of this result must be weighed carefully. While ostensibly beneficial, this artificially maintained sample size can be both a great strength and a key weakness of imputation-based approaches.

At this juncture, it is worth acknowledging that most social scientists will have only a single data set with which to test their hypotheses. So, it is unlikely that applied researchers will have much direct interest in the empirical convergence rates derived from this study. However, this result is not trivial. Higher rates of convergence observed in a Monte Carlo simulation directly translate to higher probabilities of model convergence in a *one-off* substantive study. Thus, the poor showing of the FIML conditions observed here suggests that researchers can anticipate a higher probability of achieving convergence if they treat their missing data with the SM technique rather than employing FIML estimation. This will be especially true if they are faced with small sample sizes and high rates of nonresponse.

Unfortunately, in terms of assessing direct model fit to the data, the performance of the SM technique fell far short of FIML estimation. As expected, the FIML-based estimates were very

accurate across all levels of percent missing and sample size[1]. This result should not come as a surprise and replicates the findings of previous work assessing the performance of FIML estimation (e.g., Enders & Bandalos, 2001). However, contrary to our hypothesis, the SM technique performed quite poorly when tasked with assessing direct model fit. This was particularly true when model fit was quantified by the χ^2, RMSEA, or SRMR. Indeed, other than those conditions with trivially small percents missing (i.e., $pm < 10\%$), there were no conditions tested in which the SM-based χ^2, RMSEA, or SRMR were deemed trustworthy metrics by which to judge the adequacy of the analysis models' fit to the data. However, the SM-based CFI and TLI were much more accurate, at least once sample sizes increased outside of the range that would be considered relatively small for covariance structures modeling (i.e., $N > 200$).

Although initially perplexing, this discrepancy is not entirely unintuitive when one considers the unique nature of the hypothesis tested by incremental fit indices. It must be noted that the χ^2, RMSEA, and SRMR are all derived from comparing a hypothetical, model-implied covariance matrix (which describes the modeler's expectations) to an observed covariance matrix (which describes the true response pattern in the data). Thus, these three indices are designed to test the degree to which the proposed model can replicate the observed responses. On the other hand, the CFI and TLI are both derived by comparing functions of the χ^2 values from a "worst-case" null model (which is designed to describe the data as badly as possible by ignoring any latent structure) to a similar function derived from the χ^2 of the proposed model (which describes the modeler's theory-driven expectations). Therefore, these two indices represent the degree to which a proposed model is an improvement over the worst possible explanation of the observed relationships. Clearly, the critical difference lies in the absence of the *observed* covariance matrix in the incremental approach. Because the incremental indices employ no term to quantify the raw, unstructured relationships in the data, they completely ignore the degree to which the researchers expected relationships replicate those in the observed data. Rather, these indices focus only on

[1]While the reader will note that the performance of some of the FIML-based fit indices drops off in those conditions that cross small sample sizes and high rates of missingness, these results must be interpreted in the context of the very low convergence rates for the FIML models in these conditions.

how much better a given model is than the worst possible model that can be ascribed to the same data set.

Because multiple imputation is designed to incorporate the researcher's uncertainty in the true values of the imputed data, it follows that the final imputed data sets, in some sense, represent the true population values less correctly (at least in terms of larger variances) than the complete data would have. Therefore, the discrepancies between an idealized model-implied covariance matrix and a matrix derived from imputed data should be larger than those between that same model-implied covariance matrix and a matrix derived from complete data. The χ^2, RMSEA, and SRMR would, in turn, reflect these inflated discrepancies. However, because they are impartial to the absolute discrepancies between observed and model implied covariance matrices, the CFI and TLI would not reflect the additional "badness" introduced by multiple imputation. Keeping this distinction in mind and referring back to the SM-based CFI and TLI, it is more clear why these two indices demonstrate reasonable performance when the χ^2, RMSEA, and SRMR do not. The CFI and TLI are ignoring additional misfit introduced by the missing data treatment, while the χ^2, RMSEA, and SRMR capture not only any inappropriateness of the hypothesized model but also the additional misfit that is caused by the imputation process.

Unfortunately, the positive performance of the SM-based CFI and TLI is not enough to validate its utility as a method of assessing direct model fit. While it would be possible to rely on the CFI and TLI alone to get an accurate assessment of the adequacy of your model, the responsible researcher will recognize that model fit must be viewed as a gestalt. By relying on only incremental fit indices to judge the adequacy of a given model, the modeler will get only an incomplete and possibly skewed idea of that model's appropriateness. Adding to this concern is the unique hypothesis tested by incremental fit indices. If one were to rely only upon the CFI and TLI to assess their model's accuracy, they would be ignoring all information regarding the discrepancy between the relationships implied by their hypothesized model and those truly represented by the observed data. Thus, the SM technique cannot yet be recommended as a tool to assess the direct fit of a hypothesized model to an observed sample. At the SM technique's current stage of development,

FIML estimation is still a superior tool to assess direct model fit.

Happily, the performance of the SM technique as a tool with which to conduct $\Delta\chi^2$ testing was much more promising than its performance in assessing direct model fit. In all conditions tested (except for a few condition with very high rates of missing and small sample sizes), the SM-based $\Delta\chi^2$ values showed negligible bias. This result suggests that the SM technique can produce unbiased assessments of parameter significance without the need to consider the usual assumptions regarding standard errors (e.g., distribution of the indicators, method of scale setting, correction for multiple imputations).

While unbiased SM-based $\Delta\chi^2$ tests were hypothesized and expected, an interested result that was not anticipated was the considerably poor performance of the FIML-based $\Delta\chi^2$ tests. Indeed, in all but those conditions with the lowest percents missing (i.e., $pm < 10\%$), the FIML-based $\Delta\chi^2$ values were unacceptably negatively biased. Although the effect size associated with this test (i.e., $\psi_{2,1} = r = .5$) was too large to assess rejection rates, the substantial degree of negative bias suggests that researchers seeking to capture a small effect size with FIML-based $\Delta\chi^2$ tests may face considerably inflated Type II error rates.

4.0.1 Limitations and Future Directions

Conclusions regarding the utility of the SM technique must be drawn tentatively. The scope of this study was very limited and therefore represents only a small subset of the possible modeling situations for which the applied researcher may considering using the SM technique. With regards to the data structure, all indicators where multivariate normally distributed, and the data generating and analysis models represented trivially simple, cross-sectional, single group models. None of these characteristics are likely to be the reality experienced by the applied researcher, and variation in model complexity and adherence to model assumptions could have a non-negligible impact on the performance of the techniques under study.

Future extensions of this work must address some of the important shortcomings that limit the generalizability of the current study. First and foremost, the SM technique must be compared to

additional missing data treatments, especially those that are currently recommended in the methodological community. To that end, additional simulations must be run with the Expectation Maximization (EM) Algorithm, Yuan & Bentler Two-Stage Estimator, and various corrections to the χ^2 (e.g., the Satorra-Bentler Robust χ^2) included as comparison conditions. Also, the current study was not designed to assess the accuracy of model parameters or their associated standard errors, so no conclusions can be drawn about the correctness of parameter estimates or the plausibility of hypothesis testing with Wald statistics. Considering that model fit will only be of tertiary interest to most of those who would implement the SM technique, this is a considerable shortcoming of the current work. The work of previous authors (e.g., Rubin, 1987; Satorra & Bentler, 1994; van Buuren, 2012; Wood et al., 2008) and the principles of point estimation that motivate *Rubin's Rules* would seem to suggest that the SM technique should produce unbiased point estimates of model parameters, but these suppositions must be confirmed. Future work must empirically examine the degree to which the SM technique can produce accurate parameter estimates and standard errors. Also, future work must examine the power and Type I error rates associated with the SM-based $\Delta\chi^2$. Finally, an effort must be made to address the unacceptable estimates of direct model fit. Past work has proposed several corrections to the χ^2 statistic which could be applicable to the current situation (e.g., Browne, 1984; Cai & Lee, 2009; Lee & Cai, 2012; Yuan & Bentler, 2000). Although converting the SM technique into a two stage estimator is not an optimal outcome, such corrections are likely to be the only recourse for correcting the considerable shortcomings of the SM technique as a tool to assess direct model fit.

4.1 Conclusion

In summary, the SM technique may not yet demonstrate sufficient performance to act as an "all-in-one" missing data tool. As it is currently implemented, the SM-technique does not have the capability to provide an accurate picture of overall direct model fit to the data, and this constitutes a considerable shortcoming that will keep many researchers from using the technique. However, there is much promise in the SM technique. It did outperform the Naive approach on all counts, and

when compared to FIML estimation it demonstrated far superior convergence rates. This finding suggests that researchers who employ the SM technique can expect a higher probability of reaching convergence for a given model than they can if using FIML estimation. The SM technique also outperformed FIML estimation in the context of nested model comparisons. SM-based significance tests with the $\Delta\chi^2$ demonstrated considerably more accurate conclusions than their FIML-based counterparts. These two findings in concert suggest that the SM technique will allow researchers to do more with what they have, so to speak, and offer a higher chance of observing those effects that truly exist. In short, even though the SM technique may not be ready to stand on its own quite yet, with some further modification to address its shortcomings in assessing direct model fit, it shows the potential to grow into a powerful tool in the arsenal of the applied researcher faced with the need for a simple, yet principled, missing data treatment.

References

Asparouhov, T., & Muthén, B. (2010, July). Chi-square statistics with multiple imputation [Computer software manual]. (Technical Appendix. Muthén & Muthén: Los Angeles)

Browne, M. W. (1984). Asymptotically distribution-free methods for the analysis of covariance structures. *British Journal of Mathematical and Statistical Psychology*, *37*.

Burton, A., Altman, D. G., Royston, P., & Holder, R. L. (2006). The design of simulation studies in medical statistics. *Statistics in Medicine*, *25*(24), 4279–4292.

Cai, L., & Lee, T. (2009). Covariance structure model fit testing under missing data: An application of the supplemented em algorithm. *Multivariate Behavioral Research*, *44*(2), 281–304.

Enders, C. K. (2010). *Applied missing data analysis*. The Guilford Press.

Enders, C. K., & Bandalos, D. L. (2001). The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *8*(3), 430–457.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., & Hothorn, T. (2012). mvtnorm: Multivariate normal and t distributions [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=mvtnorm` (R package version 0.9-9992)

Graham, J. W. (2003). Adding missing-data-relevant variables to fiml-based structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *10*(1), 80–100.

Honaker, J., & King, G. (2010). What to do about missing values in time-series cross-section data. *American Journal of Political Science*, *54*(2), 561–581. Retrieved from `http://dx.doi.org/10.1111/j.1540-5907.2010.00447.x`

Honaker, J., King, G., & Blackwell, M. (2011). Amelia II: A program for missing data. *Journal of Statistical Software*, *45*(7), 1–47. Retrieved from http://www.jstatsoft.org/v45/i07/

King, G., Honaker, J., Joseph, A., & Scheve, K. (2001, March). Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American Political Science Review*, *95*(1), 49–69.

L'ecuyer, P., Simard, R., Chen, E. J., & Kelton, W. D. (2002). An object-oriented random-number package with many long streams and substreams. *Operations Research*, 1073–1075.

Lee, T., & Cai, L. (2012). Alternative multiple imputation inference for mean and covariance structure modeling. *Journal of Educational and Behavioral Statistics*, *37*(6), 675–702.

Meng, X. L., & Rubin, D. B. (1992). Performing likelihood ratio tests with multiply-imputed data sets. *Biometrika*, *79*(1), 103–111.

R Development Core Team. (2011). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from http://www.R-project.org (ISBN 3-900051-07-0)

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. Retrieved from http://www.jstatsoft.org/v48/i02/

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys* (Vol. 519). Wiley Online Library.

Satorra, A., & Bentler, P. (1994). Corrections to test statistics and standard errors in covariance structure analysis.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data* (Vol. 72). Chapman & Hall/CRC.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of state of the art. *Psychological Methods*, *7*(2), 147–177.

van Buuren, S. (2012). *Flexible imputation of missing data*. Boca Raton, FL: Chapman & Hall/CRC.

Ševčíková, H., & Rossini, A. J. (2010, June). snowft: Fault tolerant simple network of workstations [Computer software manual]. Retrieved from http://www.stat.washington.edu/hana/parallel/snowFT-doc.pdf (R package version 1.2-0)

Wood, A. M., White, I. R., & Royston, P. (2008). How should variable selection be performed with multiply imputed data? *Statistics in Medicine*, *27*(17), 3227–3246. Retrieved from http://dx.doi.org/10.1002/sim.3177

Yuan, K.-H., & Bentler, P. M. (2000). Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. *Sociological Methodology*, *30*(1), 165–200. Retrieved from http://dx.doi.org/10.1111/0081-1750.00078

Appendix A: R Code for Key Functions

Code Excerpt 1: *Function to simulate complete data*

```
simData <- function(parms)
  {
    require(mvtnorm)

    nobs <- parms$nobs
    mfK <- parms$mfK
    lenScale <- parms$lenScale

    phi <- matrix(c(1,.5,.5,1),2,2)

    lambda <- matrix(c(rep(.6,(lenScale/2)),rep(0,(lenScale/2)),
                       rep(0,(lenScale/2)),rep(.6,(lenScale/2)),
                       rep(.2,(lenScale/2)),rep(.1,(lenScale/2)),
                       rep(.05,(lenScale/2)),rep(.1,(lenScale/2))
                           ),lenScale,4)

    theta <- toeplitz(c((1-.6^2),0,0,0,mfK,0,0,0,0,mfK,0,0,0,0,
       mfK,0,0,0,0,mfK))

    eta <- rmvnorm(nobs,c(0,0),phi)

    Covs <- rmvnorm(nobs,c(0,2),matrix(c(1,0,0,3),2,2))
    preds <- cbind(eta,Covs)
    colnames(preds) <- c("F1","F2","c1","c2")

    errors <- rmvnorm(nobs,rep(0,lenScale),theta)

    dat <- preds %*% t(lambda) + errors

    newDat <- cbind(dat,preds[,3:4])
    colnames(newDat) <- c(paste("a", 1:(lenScale/2), sep=""),
```

```
        paste("b", 1:(lenScale/2), sep=""),"c1","c2")

    newDat
  }# End simData()
```

Code Excerpt 2: *Function to impose missing data*

```
makeMAR <- function(pm, dat, parms)
  {
    lenScale <- parms$lenScale
    marPred1 <- parms$marPred1
    marPred2 <- parms$marPred2

        Y <- runif(lenScale*.5,0,.25*pm)

        Z <- sample(c(Y+pm,-Y+pm),replace=F)

        fun1 <- function(x,dat) pnorm(dat,mean(dat),sd(dat)) <= x

        R1 <- sapply(Z[1:(length(Z)*.5)],fun1,dat=dat[,marPred1])
        R2 <- sapply(Z[((length(Z)*.5)+1):length(Z)],fun1,dat=dat
            [,marPred2])

        R <- cbind(cbind(R1,R2)[,sample(dim(cbind(R1,R2))[2],
            replace=F)],
                    matrix(FALSE,dim(dat)[1],(dim(dat)[2]-dim(
                        cbind(R1,R2))[2])))

        dat[R] <- NA

    dat

  }# End makeMAR()
```

Code Excerpt 3: *Function to implement imputation-based missing data treatments*

```
imputeStack <- function(dat,runNumber,parms)
    {
      require(Amelia)
```

```
    require(mice)

    miceDat <- dat
    imps <- parms$imps
    lenScale <- parms$lenScale


    applyMICE <- function(index, dat, covsIndex)
      {
        mice.impute.norm.predict(dat[,index],!is.na(dat[,index
            ]),dat[,covsIndex])
      }

miceOut <- tryCatch(sapply(c(1:lenScale), FUN=applyMICE, dat=
    miceDat, covsIndex=c((lenScale+1):dim(miceDat)[2])), error=
    function(e){NULL}, finally=list())

if(is.null(miceOut)==FALSE){miceDat[is.na(miceDat)] <- unlist(
    miceOut)} else {miceDat <- NA}

    if(is.null(miceDat)==FALSE & sum(is.na(miceDat))==0){
        miceConverge <- TRUE} else {miceConverge <- FALSE;
        miceDat <- NA}

    ameliaOut <- tryCatch(amelia(dat, m=imps, empri=.1*dim(dat)
        [1], p2s=0, incheck=FALSE), error=function(e){NULL},
        finally=list())

    if(is.null(ameliaOut)==FALSE)
      {
        ameliaConverge <- TRUE

        stackedDat <- do.call("rbind", ameliaOut[[1]])

        meanImps <- matrix(apply(matrix(unlist(ameliaOut[[1]]),
            ncol=dim(dat)[1]*dim(dat)[2], byrow=TRUE), 2, mean
            ), ncol=dim(dat)[2], dimnames=list(NULL, colnames(
            dat)))

        rawCov <- lapply(ameliaOut[[1]], cov)

        superMat <- cov(stackedDat[ , 1:lenScale])
      } else {ameliaConverge <- FALSE; rawCov <- NA; superMat
        <- NA; meanImps <- NA}
```

```
    rm(stackedDat)
    rm(ameliaOut)
    rm(miceOut)

    impDat <- list(ameliaConverge=ameliaConverge, miceConverge=
        miceConverge, incompDat=dat, rawCov=rawCov, superMat=
        superMat, avImpsDat=meanImps, regImpDat=miceDat)

} # end imputeStack()
```

Code Excerpt 4: *Function to fit the CFA models for the missing data conditions*

```
fitMissinModels <- function(dat, parms)
    {
        require(lavaan)

        check <- function(lavObj) is.null(lavObj) || is.na(sum(
            unlist(inspect(lavObj,"se")))) || min(unlist(
            inspect(lavObj,"se"))) < 0

        mod1 <- parms$mod1
        mod2 <- parms$mod2
        mod3 <- parms$mod3
        nobs <- parms$nobs
        imps <- parms$imps
        lenScale <- parms$lenScale

     if(dat$ameliaConverge==TRUE)
        {
          smFullMod <- tryCatch(cfa(mod1, sample.cov=dat$superMat
              , sample.nobs=nobs, std.lv=TRUE), error=function(e)
              {NULL}, finally=list())

          smResMod <- tryCatch(cfa(mod1, sample.cov=dat$superMat,
               sample.nobs=nobs, std.lv=TRUE, orthogonal=TRUE),
              error=function(e){NULL}, finally=list())

          if(check(smFullMod)==FALSE & check(smResMod)==FALSE) {
              smConverge <- TRUE; smFullFit <- fitMeasures(
              smFullMod); smResFit <- fitMeasures(smResMod)} else
               {smConverge <- FALSE; smFullFit <- NA; smResFit <-
               NA}
```

```
avImpsFullMod <- tryCatch(cfa(mod1, sample.cov=cov(dat$
    avImpsDat[,1:lenScale]), sample.nobs=nobs, std.lv=
    TRUE), error=function(e){NULL}, finally=list())

avImpsResMod <- tryCatch(cfa(mod1, sample.cov=cov(dat$
    avImpsDat[,1:lenScale]), sample.nobs=nobs, std.lv=
    TRUE, orthogonal=TRUE), error=function(e){NULL},
    finally=list())

if(check(avImpsFullMod)==FALSE & check(avImpsResMod)==
    FALSE) {avImpsConverge <- TRUE; avImpsFullFit <-
    fitMeasures(avImpsFullMod); avImpsResFit <-
    fitMeasures(avImpsResMod)} else {avImpsConverge <-
    FALSE; avImpsFullFit <- NA; avImpsResFit <- NA}

fitRawMods <- function(x, parms)
  {
    mod1 <- parms$mod1
    nobs <- parms$nobs

    fullMod <- tryCatch(cfa(mod1, sample.cov=x, sample.
        nobs=nobs, std.lv=TRUE), error=function(e){NULL
        }, finally=list())

    resMod <- tryCatch(cfa(mod1, sample.cov=x, sample.
        nobs=nobs, std.lv=TRUE, orthogonal=TRUE), error
        =function(e){NULL}, finally=list())

    if(check(fullMod)==FALSE & check(resMod)==FALSE)
      {
        rawConverge <- TRUE
        list(rawConverge=rawConverge, rawFullFit=
            fitMeasures(fullMod),rawResFit=fitMeasures(
            resMod))
      } else {rawConverge <- FALSE; list(rawConverge=
          rawConverge)}
  }

rawMissOut <- lapply(dat$rawCov,FUN=fitRawMods,parms=
    parms)

convergeList <- list()
for(i in 1:length(dat$rawCov)) convergeList[[i]] <-
    rawMissOut[[i]]$rawConverge
```

```
    if(sum(unlist(convergeList)) == length(dat$rawCov))
      {
        naiveConverge <- TRUE

        naiveFit <- apply(matrix(unlist(rawMissOut), length
            (rawMissOut), 41, byrow=TRUE, dimnames=list(
            NULL,c("rawConverge",rep(labels(rawMissOut[[1]]
            $rawFullFit),2)))), 2, FUN=mean)

        naiveFullFit <- naiveFit[2:21]; naiveResFit <-
            naiveFit[22:41]

      } else {naiveConverge <- FALSE; naiveFullFit <- NA;
         naiveResFit <- NA}

  } else {smConverge <- NA; naiveConverge <- NA;
    avImpsConverge <- NA; smFullFit <- NA; smResFit <- NA
    ; naiveFullFit <- NA; naiveResFit <- NA;
    avImpsFullFit <- NA; avImpsResFit <- NA}


if(dat$miceConverge==TRUE)
  {
    regImpFullMod <- tryCatch(cfa(mod1, sample.cov=cov(dat$
        regImpDat[,1:lenScale]), sample.nobs=nobs, std.lv=
        TRUE), error=function(e){NULL}, finally=list())

    regImpResMod <- tryCatch(cfa(mod1, sample.cov=cov(dat$
        regImpDat[,1:lenScale]), sample.nobs=nobs, std.lv=
        TRUE, orthogonal=TRUE), error=function(e){NULL},
        finally=list())

    if(check(regImpFullMod)==FALSE & check(regImpResMod)==
        FALSE)
      {regImpConverge <- TRUE; regImpFullFit <- fitMeasures
         (regImpFullMod); regImpResFit <- fitMeasures(
         regImpResMod)} else {regImpConverge <- FALSE;
         regImpFullFit <- NA; regImpResFit <- NA}

  } else {regImpConverge <- NA; regImpFullFit <- NA;
    regImpResFit <- NA}

    fimlNullMod <- tryCatch(cfa(mod3, data=as.data.frame(
        dat$incompDat), missing="FIML"), error=function(e){
```

```
    NULL}, finally=list())

fimlFullMod <- tryCatch(cfa(mod2, data=as.data.frame(
    dat$incompDat), missing="FIML", std.lv=TRUE), error
    =function(e){NULL}, finally=list())

fimlResMod <- tryCatch(cfa(mod2, data=as.data.frame(dat
    $incompDat), missing="FIML", std.lv=TRUE,
    orthogonal=TRUE), error=function(e){NULL}, finally=
    list())

if(check(fimlFullMod)==FALSE & check(fimlResMod)==FALSE
    & check(fimlNullMod)==FALSE)
  {
    fimlConverge <- TRUE

    fimlFullCFI <- max(0,(fitMeasures(fimlNullMod,"
        chisq")-fitMeasures(fimlNullMod,"df"))-(
        fitMeasures(fimlFullMod,"chisq")-fitMeasures(
        fimlFullMod,"df")))/(fitMeasures(fimlNullMod,"
        chisq")-fitMeasures(fimlNullMod,"df"))

    fimlResCFI <- max(0,(fitMeasures(fimlNullMod,"chisq
        ")-fitMeasures(fimlNullMod,"df"))-(fitMeasures(
        fimlResMod,"chisq")-fitMeasures(fimlResMod,"df"
        )))/(fitMeasures(fimlNullMod,"chisq")-
        fitMeasures(fimlNullMod,"df"))

    fimlFullTLI <- ((fitMeasures(fimlNullMod,"chisq")/
        fitMeasures(fimlNullMod,"df"))-(fitMeasures(
        fimlFullMod,"chisq")/fitMeasures(fimlFullMod,"
        df")))/((fitMeasures(fimlNullMod,"chisq")/
        fitMeasures(fimlNullMod,"df"))-1)

    fimlResTLI <- ((fitMeasures(fimlNullMod,"chisq")/
        fitMeasures(fimlNullMod,"df"))-(fitMeasures(
        fimlResMod,"chisq")/fitMeasures(fimlResMod,"df"
        )))/((fitMeasures(fimlNullMod,"chisq")/
        fitMeasures(fimlNullMod,"df"))-1)

    fimlFullFit <- c(fitMeasures(fimlFullMod)[1:6],
        fimlFullCFI,fimlFullTLI,fitMeasures(fimlFullMod
        )[9:20])
    names(fimlFullFit) <- labels(fitMeasures(
        fimlFullMod))
```

```
            fimlResFit <- c(fitMeasures(fimlResMod)[1:6],
                fimlResCFI, fimlResTLI, fitMeasures(fimlResMod)
                [9:20])
            names(fimlResFit) <- labels(fitMeasures(fimlResMod)
                )

        } else {fimlConverge <- FALSE; fimlFullFit <- NA;
            fimlResFit <- NA}

    converge <- matrix(c(dat$ameliaConverge, dat$
        miceConverge, fimlConverge, smConverge,
        naiveConverge, avImpsConverge, regImpConverge),
        ncol=7, dimnames=list(NULL, c("amelia","mice","fiml"
        ,"sm","naive","avImps","regImp")))

    list(converge=converge, fimlFullFit=fimlFullFit,
        fimlResFit=fimlResFit, smFullFit=smFullFit,
        smResFit=smResFit, naiveFullFit=naiveFullFit,
        naiveResFit=naiveResFit, avImpsFullFit=
        avImpsFullFit, avImpsResFit=avImpsResFit,
        regImpFullFit=regImpFullFit, regImpResFit=
        regImpResFit)

}# End fitMissinModels()
```